

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

PERFORMANCE TESTING FOR THE
MARINE AIR GROUND TASK FORCE
TACTICAL WARFARE SIMULATION

by

William Alan Sawyers

September 1995

Thesis Advisor:

William G. Kemple

Approved for public release; distribution is unlimited

19960220 049

DTIC QUALITY INSPECTED 1

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (<i>Leave blank</i>)	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE PERFORMANCE TESTING FOR THE MARINE AIR GROUND TASK FORCE TACTICAL WARFARE SIMULATION			5. FUNDING NUMBERS
6. AUTHOR(S) Sawyers, William A.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (<i>maximum 200 words</i>) The Marine Air Ground Task Force MAGTF) Tactical Warfare Simulation (MTWS) is a computer-assisted wargame being developed to provide a cost effective, yet realistic, training environment for Marine commanders and their staffs. A Developmental Test, conducted in November 1994, highlighted the need to improve the overall performance of the system. However, performance testing methods, which were used to evaluate the timeliness of events and the responsiveness of the simulation processes, were relatively new and unproven. A more thorough analysis of MTWS Developmental Test data and performance testing techniques should provide valuable insight for suggesting improvements. With this purpose in mind, this thesis conducts a detailed analysis of the MTWS Developmental Test to assess the statistical significance of the test results, recommend improved performance measures, establish a quantifiable baseline for evaluating future MTWS configurations, and recommend enhanced testing procedures for assessing performance. Since performance testing will continue throughout the system's life cycle, it is hoped that many of these suggestions and techniques will be adopted in subsequent tests. Much of this insight may apply not only to MTWS, but to other wargaming systems as well. Broad issues relating to system performance are discussed in terms of the specification, design, and testing of computer-based warfare simulations.			
14. SUBJECT TERMS Performance Testing, Tactical Warfare Simulation, Measures of Performance, Simulation Testing Procedures			15. NUMBER OF PAGES 99
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

Approved for public release; distribution is unlimited.

**PERFORMANCE TESTING FOR THE
MARINE AIR GROUND TASK FORCE
TACTICAL WARFARE SIMULATION**

William A. Sawyers
Major, United States Marine Corps
A.B., Duke University, 1978
M.S., University of Southern California, 1983
M.S., University of Idaho, 1986

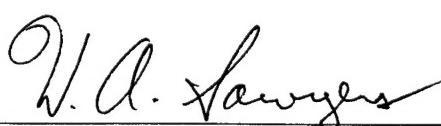
Submitted in partial fulfillment
of the requirements for the degree

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

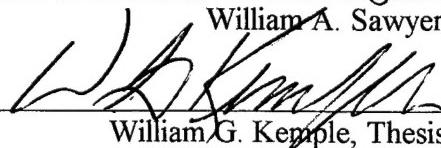
NAVAL POSTGRADUATE SCHOOL
September 1995

Author:

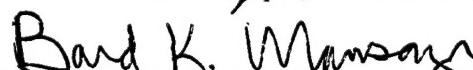


William A. Sawyers

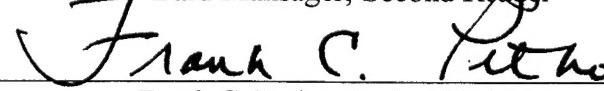
Approved by:



William G. Kemple, Thesis Advisor



Bard K. Mansager, Second Reader



Frank C. Petho, Acting Chairman
Department of Operations Research

ABSTRACT

The Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS) is a computer-assisted wargame being developed to provide a cost effective, yet realistic, training environment for Marine commanders and their staffs. A Developmental Test, conducted in November 1994, highlighted the need to improve the overall performance of the system. However, performance testing methods, which were used to evaluate the timeliness of events and the responsiveness of the simulation processes, were relatively new and unproven. A more thorough analysis of MTWS Developmental Test data and performance testing techniques should provide valuable insight for suggesting improvements.

With this purpose in mind, this thesis conducts a detailed analysis of the MTWS Developmental Test to assess the statistical significance of the test results, recommend improved performance measures, establish a quantifiable baseline for evaluating future MTWS configurations, and recommend enhanced testing procedures for assessing performance. Since performance testing will continue throughout the system's life cycle, it is hoped that many of these suggestions and techniques will be adopted in subsequent tests.

Much of this insight may apply not only to MTWS, but to other wargaming systems as well. Broad issues relating to system performance are discussed in terms of the specification, design, and testing of computer-based warfare simulations.

TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. PROBLEM	2
C. OBJECTIVES	2
D. THESIS OVERVIEW	3
II. MTWS PROGRAM AND SYSTEM DESIGN	5
A. PURPOSE AND EMPLOYMENT CONCEPTS	5
B. SIMULATION FUNCTIONS AND FEATURES	6
C. SYSTEM DESIGN AND ARCHITECTURE	8
1. Design Philosophy	8
2. Software Configuration	8
3. Hardware Configuration	12
III. MTWS DEVELOPMENTAL TEST	15
A. OVERVIEW AND OBJECTIVES	15
B. TIMING PERFORMANCE MEASURES	17
C. TEST CONFIGURATION	18
D. DATA COLLECTION AND ANALYSIS	20
E. TEST RESULTS	22
IV. POST DEVELOPMENTAL TEST ANALYSIS	25
A. CRITIQUE OF DEVELOPMENTAL TEST	26
B. DATA COLLECTION AND REDUCTION	27
C. ASSESSMENT OF DEVELOPMENTAL TEST RESULTS	29
1. Event Timing	29

2. Cycle Length Sampling	33
D. DETAILED ANALYSIS	40
1. Intelligence Cycle	40
2. Ground Combat Cycle	44
3. Event Time Differential	47
E. DISCUSSION	54
V. RECOMMENDATIONS	59
A. MEASURES OF PERFORMANCE	59
B. DATA COLLECTION	64
C. PERFORMANCE TESTING SCENARIO	65
D. PERFORMANCE BASELINE	68
E. LESSONS LEARNED	70
1. Specification	71
2. Design	71
3. Testing	72
VI. CONCLUSIONS	73
APPENDIX. GRAPHICAL ANALYSIS OF DIFFERENT MULTINOMIAL PROBABILITIES	75
LIST OF REFERENCES	81
INITIAL DISTRIBUTION	85

EXECUTIVE SUMMARY

The performance of tactical warfare simulations has become a critical issue as the scope and complexity of these systems have dramatically grown. Computer-assisted wargames have become increasingly important as cost effective tools for training military commanders and staffs. Although much effort is usually devoted to defining what such models must simulate, system specifications seldom address how well the system must perform these desired functions. In this context, performance refers to the timeliness of events and the responsiveness of the simulation processes. When performance lags, the training value of a wargame is diminished.

This paper closely examines the performance of the Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS), a wargame recently developed for Marine Corps training and scheduled for fielding in the second half of FY95. A Developmental Test, conducted in November 1994, highlighted the need to improve the overall performance of the system. The Developmental Test represented the first attempt to assess MTWS system performance in detail. As such, many of the data collection, analysis, and testing techniques were new and unproven. This thesis conducts a detailed analysis of the MTWS Developmental Test to assess the statistical significance of the test results, recommend improved performance measures, establish a quantifiable baseline for

evaluating future MTWS configurations, and recommend enhanced testing procedures for evaluating performance.

Data for this study were extracted from computer printouts and archived files generated during the MTWS Developmental Test. The analysis focuses on three main areas of performance: 1) the timeliness of scenario events; 2) the run-time efficiency of the intelligence algorithm; and 3) the run-time efficiency of the ground combat algorithm.

The original test report, published in December 1994, stated that timing problems were most evident for ground movements; this finding is shown to be statistically significant (p -value < 0.0001). Additionally, graphical analysis reveals the need to develop improved data collection methods for gathering MTWS run-time data, and automated data collection techniques are recommended to save time while also ensuring the overall accuracy of the data.

Measures of performance (MOPs) are developed to reflect both statistical and operational considerations. These benchmarks will facilitate the statistical comparison between various MTWS configurations to determine if performance has been significantly improved or altered in subsequent releases. A total of 35 measures covering seven areas of interest are proposed. Since these measures are derived directly from ratio-scaled data, more powerful statistical tests can be employed than with the ordinal-based data originally gathered during the Developmental Test. A quantifiable baseline is then produced by gathering data for these MOPs over the portion of the test scenario exhibiting peak computational load.

Since performance testing will continue throughout the system's life cycle, it is hoped that many of these suggestions and techniques will be adopted in subsequent tests. As the size and complexity of the simulation increases, performance of MTWS will continue to be a concern. Several lessons may apply not only to MTWS, but to other wargaming systems as well. Broad issues relating to system performance are discussed in terms of the specification, design, and testing of computer-based warfare simulations.

I. INTRODUCTION

A. BACKGROUND

The Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS) is a recently developed, computer-assisted wargame. MTWS is designed to provide a cost effective, yet realistic, training environment for Marine commanders and their staffs well into the 21st century. It will also provide the means for the Marine Corps to participate actively in joint gaming exercises, a critical capability that is now lacking [Ref. 1]. Current plans call for fielding MTWS in the later half of fiscal year 95 upon completing a series of tests.

The first test was a formal Developmental Test conducted at Camp Pendleton, CA on 14 - 19 November 1994. The test objective was to demonstrate the capability of MTWS to support a Marine Expeditionary Force (MEF) level exercise [Ref. 2]. Although substantial progress was demonstrated, the test highlighted the need to improve the overall performance of the system. Performance is defined as the timeliness of events and responsiveness of the simulation processes. When performance lags, the training value of the simulation is diminished. As a result of the Developmental Test, enhancing system performance was identified as the most critical concern facing MTWS for fielding [Ref. 3].

The Developmental Test was the first attempt to assess MTWS system performance in detail. As such, many of the data collection, analysis, and testing

techniques were new and unproven. Although much was learned from the test, further insight can be gained by conducting a more thorough analysis and developing more precise performance measures. This will greatly assist in establishing an accurate baseline to be used in charting the performance of future releases. A comprehensive review of the Developmental Test methodology, data, and results should yield many valuable lessons. Since performance testing will continue throughout the life cycle of the system, these lessons can be applied many times in subsequent tests.

B. PROBLEM

To resolve these issues, performance measures should be refined to provide a suitable framework for future testing of MTWS. Data collection methods need to be simplified and testing procedures should be improved to support more timely and meaningful analysis. A performance baseline needs to be established so the project management office can ensure that the system delivers realistic play and meets Marine Corps requirements.

C. OBJECTIVES

As part of this study, the author participated in the MTWS Developmental Test. The thesis will discuss the test and conduct detailed post test analysis to:

1. Identify trends and/or relationships regarding system performance;
2. Recommend improved performance measures and establish a quantifiable performance baseline for evaluating future software and hardware configurations;

3. Recommend improved testing procedures for assessing MTWS performance throughout the life cycle of the system; and
4. Discuss aspects of performance specification, design, and testing that can be applied to war-gaming systems in general.

D. THESIS OVERVIEW

The remainder of the thesis is organized as follows:

- ◆ Chapter II provides an overview of the MTWS program and system design.
- ◆ Chapter III discusses the Developmental Test procedures and results.
- ◆ Chapter IV contains detailed analysis of data collected both during and after the Developmental Test.
- ◆ Chapter V presents specific recommendations for improving MTWS performance testing and discusses lessons learned that may apply to other wargaming systems.
- ◆ Chapter VI briefly reviews the conclusions of the thesis.

II. MTWS PROGRAM AND SYSTEM DESIGN

A. PURPOSE AND EMPLOYMENT CONCEPTS

The primary purpose of the MTWS program is to enhance training of tactical commanders and their staffs for Fleet Marine Force (FMF) units and selected Marine Corps schools. MTWS will normally be used to support command post exercises (CPXs) in which combat forces, supporting arms, and results of combat are all simulated by the system. In this role, the system will be the primary tool of the exercise controllers, who are usually members of the tactical exercise control group. Throughout a CPX, MTWS is used to exercise the gamut of command and staff functions, in near-real-time, from battalion through MEF level [Ref. 4]. This challenging requirement demands detailed, yet efficient, algorithms coupled with computer hardware of great computational speed and capacity. MTWS can also support Field Exercises (FEXs) in which all or part of the forces are actual units exercising in the field. In FEX play, the system is used to record and monitor the actions of the live forces rather than simulating such actions; MTWS can also be used to adjudicate simulated conflicts in war games involving real maneuver forces [Ref. 5].

A significant role as an analytic tool is also envisioned for MTWS. Since MTWS is extremely transportable, it can deploy with Marine units to the area of operations. Thus, Marines can use MTWS on a tactical level to assist in planning actual operations by gaming alternate courses of action. Once a concept of operation is determined, MTWS

can be used to refine the plan under various conditions; contingency plans can be similarly tested and rehearsed. Looking to the future, the Marine Corps should also be able to assess the impact of proposed weapon systems or proposed doctrinal changes using MTWS. As defense budgets continue to shrink, the importance of MTWS to the Marine Corps will continue to grow as a cost effective means for conducting realistic combat training and analysis.

B. SIMULATION FUNCTIONS AND FEATURES

Curtis L. Blais, the Software Engineering Manager for MTWS, best summarizes the capabilities of the system when he states [Ref. 6],

MTWS provides a full spectrum of combat models required to simulate Marine tactical exercises. The major functional areas are Ground Combat, Air Operations, Fire Support, Ship to Shore, Combat Service Support, Combat Engineering, and Intelligence. The system provides limited play in Electronic Warfare, Communications, and Nuclear, Biological, and Chemical Warfare.

The Tactical Warfare Simulation Evaluation and Analysis System (TWSEAS), an aging computer based simulation widely employed by the Marine Corps, does not support such a wide range of battlefield activities and has reached obsolescence. No single Department of Defense (DoD) combat simulation is capable of modeling battle on land, sea, and air to the degree of detail required by Marines. The ability to faithfully replicate the equipment, organization, doctrine, tactics, and techniques of Marine units from battalion through MEF levels distinguishes MTWS from other existing simulations. This is particularly vital in the area of amphibious operations.

MTWS imports Digital Terrain Elevation Data and Digital Feature Analysis Data from the Defense Mapping Agency. This provides a ready-made database of trafficability, vegetation, cover, and elevation information virtually anywhere in the world. MTWS users can also enter user defined terrain features, obstacles, and weather conditions. MTWS models account for these factors when simulating movements and detections. Up to four million terrain data points can be stored in the system. This permits coverage of a 200 x 200 kilometer area with terrain resolution of 100 meters or up to a 1000 x 1000 kilometer area with terrain resolution of 500 meters [Ref. 7].

MTWS reports information to the user in two distinct formats, solicited reports and spot reports. Solicited reports are pre-formatted queries of the exercise database which can be initiated by the user. There are a wide variety of solicited reports, and these can be tailored by defining filters for displaying a specific subset of the available data. Spot reports are generated automatically by the combat simulation models to inform operators of all relevant battlefield developments. These include such matters as enemy detections, unit actions, battle damage assessments, and casualties incurred from combat. The stream of spot reports and the map display keep the operator well informed as to the tactical situation. Additional, more detailed information is provided through solicited reports when required. All reports are labeled with game-time rounded to the nearest minute to form a chronological record of the battle.

C. SYSTEM DESIGN AND ARCHITECTURE

1. Design Philosophy

MTWS is hosted on a distributed network of UNIX workstations. This design provides a flexible, robust, and highly portable system architecture. The fundamental design philosophy is that "the controller drives the game, not the simulation software" [Ref. 8]. Thus, although MTWS attempts to make reasonable tactical decisions to relieve operators of low-level management tasks, controllers can always override any automated decisions. In fact, controllers can manually input detailed commands to control every aspect of unit actions if desired. Usually, controllers will rely on the discretion of the system for convenience, but selectively override certain responses.

The MTWS program is deeply committed to meeting all applicable Department of Defense (DoD) standards as well as widely accepted software development practices. The overall goal is to build well-documented software based upon an open system architecture that will be easy to maintain and enhance. This design will accommodate significant growth which is envisioned for MTWS over its life cycle.

2. Software Configuration

The MTWS software consists of three Computer Software Configuration Items (CSCIs): 1) the MTWS Application Network (MAN); 2) the MTWS System Control (MSC); and 3) the MTWS Display System (MDS). The MAN contains the combat models and algorithms that conduct the battlefield simulation, control the exercise database, and generate spot reports. The MSC provides overall control and

synchronization to all stations of the MTWS network. The MSC manages game-time, routes commands for processing, initiates the generation of solicited reports, and conducts all system administration functions. The MDS provides the user interface to include command entry, map display, and report presentation functions [Ref. 9]. These CSCIs are discussed in more detail below.

The MAN CSCI is the heart of the simulation. It contains the intelligence and ground combat algorithms, two of the primary functions of the system. The intelligence algorithm determines detections between units and objects in the database by means of visual, aural, or ground sensor assets. The ground combat algorithm simulates battle between ground units. It progressively determines the outcomes of conflicts and assesses casualties. Both algorithms have been metered with a time stamp routine so that the actual performance of the system can be precisely measured in terms of run-time. A message is spooled to the appropriate MAN console stating the cycle number, cycle length, and time of the reading. Both processes run virtually concurrent when MTWS is operating. The meaning of these cycle length readings is discussed in the following paragraphs.

The Intelligence (IN) Cycle is the time in seconds to complete a pass through the intelligence algorithm code before looping back to the beginning of the process. The IN Cycle directly measures the elapsed run-time for MTWS to simulate intelligence functions for collection assets and units. During a cycle, the intelligence algorithm updates detection relationship between all ground sensors and all units. If changes occur, appropriate spot

reports are initiated. Intelligence processing is constantly running throughout an exercise to determine the type and extent of knowledge between forces based on detection probabilities. Longer IN Cycle lengths indicate periods when high computational demands are placed on the intelligence algorithm. High IN Cycle values reflect that more time was required to complete detection processing due to a variety of factors in the tactical situation.

Similarly, the Ground Combat (GC) Cycle is the time in seconds for the ground combat algorithm to complete a pass through its code before looping back to the beginning. The GC Cycle measures elapsed run-time for MTWS to simulate ground combat functions such as threat evaluations, unit strength assessments, and engagement updates based on the tactical situation. During a cycle, the ground combat algorithm updates the exercise database, such as effective personnel strength, weapon status, ammunition counts, etc., and initiates appropriate spot reports to reflect the results of combat. Ground combat processing is constantly running during an exercise. Longer GC Cycle lengths indicate periods when greater stress is placed on the ground combat algorithm; this usually occurs when the frequency of enemy units detected within direct fire range increases. High GC readings reflect that more time was required to complete ground combat processing for a given tactical situation.

The MSC acts as the brain of MTWS by coordinating the activities of numerous concurrent processes. It is primarily responsible for managing the game-time for MTWS and sending time updates throughout the distributed network. Due to the emphasis in the

system specification for developing a near-real-time simulation, MTWS employs a unique time management scheme. Game-time is not coupled to the processing of events in the events list as is the case in discrete event simulations; rather the MSC advances time independent of events but according to the desired speed of the game. The user can specify game-time from as slow as 1/8 to as fast as 10 times real time. Thus, all times listed on MTWS reports reflect the actual game-time according the desired speed of the scenario.

Time progresses at a steady pace even if the computational demands placed on the system exceed the capacity of the processors in the network. This provides tremendous visibility into the performance of the system; it is easy to discern whether the system is operating at the desired speed or not. For example, an event scheduled to commence at 0900Z may actually be executed at 0905Z. MTWS does not hide this fact from the user by slowing down the clock to match the processing of events as is done by most simulations. This timing lag is known as the "event time differential," which is the time in minutes between when an event was scheduled to occur and when it actually occurred in the exercise. This measure will play a key role in assessing the performance of the system. The scheduled game-time is specified as part of the command initiating the event, while the execution game-time is recorded on the applicable spot report generated by the event.

The drawback to the MTWS approach is that events can get out of synchronization and thus undermine the fidelity of the wargame. This can be a serious

problem. The challenge to the MTWS developer is to produce models of high run time efficiency so that the near-real-time goal can be achieved while maintaining the timing of events relative to one another.

The MTWS system software is written primarily in the Ada programming language per DoD regulations. MTWS consists of approximately 200,000 lines of Ada source code. A government-off-the-shelf map server was used for management and display of digitized maps. MTWS was developed per DoD-STD-2167A, Defense Systems Software Development, to provide comprehensive documentation of the software specification, requirements, and design [Ref. 10]. MTWS is one of the few computer-based simulations to have complied with both the Ada and 2167A requirements. Although this has been costly in terms of time and effort during development, significant cost savings should accrue over the life cycle of the system.

3. Hardware Configuration

MTWS is hosted on commercial Hewlett-Packard (HP) 9000 series workstations procured from the Navy's TAC-3 contract [Ref. 11]. The hardware configuration is based upon the CSCIs, but can vary depending upon the size and needs of the exercise. The MAN is normally hosted on three HP 750 processors, and the MSC requires another HP 750. The number of MDS workstations can vary from at least 1 to as many as 26 HP 730 processors. Thus, the MTWS network usually requires at least 5 workstations (i.e., 3 MAN, 1 MSC, and 1 MDS) but can expand up to 30 or more workstations depending on the number of displays needed for exercise controllers and the number of MAN processors

used to spread the simulation processing load. Due to the importance of the MAN and MSC components, an upgrade from HP 750 to HP 755 workstations was initiated in the first half of FY 95. This has increased the processing power of these hardware components from 76 to 124 million instructions per second.

The network is linked through a standard Ethernet connection. The MSC workstation has two Ethernet ports, one connecting the MAN workstations and the other connecting the MDS workstation(s). Thus, all interactions and data transfers between the MAN and MDS processors must pass through the MSC workstation for routing.

III. MTWS DEVELOPMENTAL TEST

A. OVERVIEW AND OBJECTIVES

As MTWS development progressed, a full scale Developmental Test was conducted at Camp Pendleton, CA from 14 - 19 November 1994. The purpose of the test was to determine the system's capabilities and shortcomings in support of a MEF-level tactical exercise. The test assessed four broad areas: 1) functionality; 2) timing; 3) capacity; and 4) reliability [Ref. 12].

Before the Developmental Test, reliability was the greatest concern of both MTWS users and developers due to frequent system crashes. However, the test provided ample proof that recent modifications to the Ada compiler as well as the MTWS software had dramatically improved the stability of the system [Ref. 13]. During the Developmental Test, the performance of the system as assessed by timing measures became the paramount concern. This thesis will focus exclusively on this issue.

The scenario for the Developmental Test involved joint operations against opposing forces (OPFOR) of the North Korean Peoples' Army in the Republic of Korea. MTWS simulated play of two carrier battle groups, an amphibious task force, two U.S. Army brigades, a MEF, and numerous Air Force aircraft and airfields. A night-time amphibious assault was conducted to land a Marine Regimental Landing Team (RLT) at H-hour in the OPFOR rear area. All friendly forces were referred to as the Landing Force (LF). More than 550 ground units were created in the exercise database along with hundreds of other

database objects (i.e., aircraft, ships, tactical control measures, targets, etc.) to support the simulation. The scenario was designed to provide at least 72 hours of continuous play [Ref. 14].

The Developmental Test was divided into four main phases: Rehearsal, Phase 1, Phase 2, and Follow-on Phase [Ref. 15]. Since performance data was collected only during Phase 1, this study will only analyze that portion of the test. Phase 1 was conducted over three consecutive days during which more than 24 hours of the scenario (H-14 through H+10) were played. The CPX type exercise was suspended each evening and resumed the following morning.

Phase 1 relied primarily on batch files to drive the game. The batch files each contained a series of pre-defined MTWS commands prepared specifically to support the Developmental Test. Terminal operators were required to enter these files into the system at predetermined times according to the master scenario list. Use of scripted batch files offers several advantages over keyboard input or "free-play" during testing. Batch files enable multiple commands to place demands upon the system almost simultaneously. This provides the stress required to conduct meaningful performance testing. The batch files also establish control and repeatability over the test scenario [Ref. 16].

The only drawback to using batch files is that they do not engage the operator's creativity and involvement. After a full day of testing, several operators began to lose interest in the game. Therefore, the test participants were granted permission to conduct limited free-play during the later two days of Phase 1 to relieve boredom [Ref. 17]. This

introduced a small source of additional variability into the data. However, the overwhelming majority of executable commands came from batch files; the operator generated input had little effect on the overall conduct or results of the test.

B. TIMING PERFORMANCE MEASURES

In general, defense systems are tested against the requirements delineated in their Operational Requirements Document (ORD) and their System/Segment Specification (SSS). In the case of MTWS, few quantifiable performance measures were specified in the baseline documentation. The ORD requires near-real-time control of exercise play and specifies that "system response will be a maximum of 5 seconds" [Ref. 18]. This was interpreted by the developer to mean that MTWS would acknowledge the receipt of commands and report requests within an average of five seconds of entry; the system must initiate appropriate action in this time-frame. However, no performance metrics were specified to govern when the activities and processes would be completed other than in "near-real-time". Although this term is a valid design goal, it is somewhat vague and does not constitute readily testable criteria by itself. Therefore, more specific timing measures were outlined in the Developmental Test Plan.

Timing was defined as the ability of the system to perform planned combat operations and exercise activities on-time to facilitate exercise control. Specific goals for the developmental test were established as follows [Ref. 19]:

1. No less than 80% of scheduled movements, air events, fire missions, and ship-to-shore events should occur within one minute of scheduled time.

2. The remaining 20% of scheduled movements, air events, fire missions, and ship-to-shore events should occur no later than two minutes after the scheduled time.

Data for calculating these percentages are derived from event time differential data determined by comparing the game-time which an event was scheduled to the game-time when it occurred. Since all MTWS reports are time stamped with a date time group accurate to minutes, timing data is rounded to the nearest minute. Events occurring between plus or minus one minute of the scheduled time were considered on-time. Events occurring beyond one minute of the scheduled time were categorized as late. The test plan did not mention any performance measures associated with the IN and GC Cycle lengths. Although IN and GC Cycle lengths directly reflect system responsiveness and are readily available, no formal plans were made to collect and analyze this data. However, just before the start of Phase 1, it was decided to record a sample of this data every three hours. This was easy to accomplish since the two cycle lengths scroll across the display window of their respective MAN terminals throughout an exercise. A total of eight readings was taken over the 24 operational hours of Phase 1.

C. TEST CONFIGURATION

An MTWS network consisting of 29 workstations was used in the Developmental Test. The test configuration is summarized in Table 1. Note that the enhanced capabilities of the HP 755 processors were not yet available for use. It is estimated that upgrading all MAN and MSC terminals to HP 755's may improve performance as much as 35% [Ref. 20].

<u>Workstation(s)</u>	<u>Functional Area(s)</u>	<u>Terminal</u>
MAN 001	Air, Ship-to-Shore, Engineering, & Fire Support Simulations	1 HP 750
MAN 002	Intelligence Simulation	1 HP 750
MAN 003	Ground Combat & Combat Service Support Simulations	1 HP 750
MSC 001	System Control & Administration	1 HP 750
MDS 001	Test Director	1 HP 730
MDS 002-005	Data Collection Cell	4 HP 730
MDS 006-007 017-018	Landing Force Air, Ship-to-Shore, & Intelligence Cell	4 HP 730
MDS 008-011	Landing Force Artillery & Logistics Cell	4 HP 730
MDS 012-015	Landing Force Maneuver Cell	4 HP 730
MDS 019-022 023-026	Aggressor Maneuver, Artillery & Air Cell	8 HP 730

**Table 1. Developmental Test Configuration
From [Ref. 21]**

The specific software configuration tested during Phase 1 is shown in Table 2 [Ref. 22]. Version ar115.4 was the developmental build used to assess MTWS performance.

Software Item	Version
Operating System	HP-UX 9.05
Map Server	1.13.2
MTWS	ar115.4

Table 2. Phase 1 Software Configuration

D. DATA COLLECTION AND ANALYSIS

The sole performance measures defined in the test plan were counts of the number of events that were on-time, two minutes late, or greater than two minutes late. All spot reports were spooled to a high speed printer to create a permanent record of the times specific exercise events occurred. The controllers of each exercise cell were required to document and report late events in their respective areas [Ref. 23]. However, it soon became evident that some controllers were more thorough than others in accomplishing this task. Rather than allowing the "human element" to influence the data, the complete data set was gathered after the Developmental Test ended.

This data collection task involved manually comparing scheduled event times listed in the command batch files to the execution times specified in the associated MTWS spot report. Determining event time differential data for all Phase 1 events required more than

two man-weeks of effort. It was necessary to search through a stack of computer printouts approximately one foot-high to locate the spot reports caused by each command.

MTWS is an aggregation of deterministic and probabilistic models, so there were some cases when an event was executed late due to the play of the game rather than computational overload. For example, a reconnaissance aircraft may be late in reaching its designated station due to taking evasive action to avoid encounters with hostile aircraft, or a unit may be slow to cross the line of departure when attacking if obstacles are encountered enroute from the assembly area. Such cases are not the result of timing problems; they are a routine part of the system's capability to simulate a real battlefield. Therefore, attempts were made to distinguish events that were late due to timing problems from those that were late due to valid operational reasons within the context of the game. Such events were dropped from the analysis because the timing could not be properly categorized as either late or on-time. Although much time and effort were expended to capture accurate time differential data, it was partly a subjective, and thus possibly imprecise, endeavor. Data collection would have been much easier if the spot reports had been spooled to a data file as well as to the printer. Basic text search utilities could then have been used to search the file for specific items of interest.

The IN and GC cycle length data recorded during Phase 1 were not analyzed. Although interesting and germane, it was felt that eight measurements taken over the 24 hours of the exercise were too few to be considered a representative sample; this decision

is examined in Chapter IV. Care was taken not to draw inferences from data that might not indicate the true performance of the system.

E. TEST RESULTS

The Developmental Test timing data is summarized in Table 3. The event time differential data for all five event types (i.e. air missions, fire missions, ground movements, ship-to-shore movements, and ship movements) were categorized into one of three possible categories. The on-time category includes all events with an event time differential between +/- one minute. The next column lists events which were recorded as exactly two minutes late. The remaining events (greater than two minutes late) were grouped into one broad category; values ranged from three to 22 minutes late. Thus, event time differential data was essentially transformed into five sets of multinomial data by incrementing a counter for the appropriate bin.

The test report stated that timing problems were most pronounced for ground movements. This is supported by Table 3, but no tests were conducted to determine the statistical significance of this observation. The most severe timing delay occurred at H+4 in the scenario when 77 ground units were directed to move simultaneously. This resulted in a 22 minute lag for some of these events. The report also observed that there seemed to be a strong correlation between the scheduling of large ground movements and the occurrence of late fire and air missions. Overall, the test highlighted the need to improve

Events	Number of Events Tested	Events On-time	Events 2 Minutes Late	Events > 2 Minutes Late	Percent On-time	Percent Late
Air Missions	201	186	10	5	92.6%	7.4%
Fire Missions	313	264	19	30	84.3%	15.7%
Ground Movements	222	30	12	180	13.5%	86.5%
Ship-to-Shore	23	23	0	0	100.0%	0.0%
Ship Movements	22	22	0	0	100.0%	0.0%
TOTAL	781	525	41	215	67.2%	32.8%

**Table 3. Developmental Test Timing Data
From Ref. [24]**

the timeliness and responsiveness of the system processes in general. Ground movements were specifically identified as the primary area of concern [Ref. 25].

IV. POST DEVELOPMENTAL TEST ANALYSIS

A few key concepts must be kept in perspective when reviewing the results of the MTWS Developmental Test. First, all complex software has defects. The objective of software testing is to find and document as many "bugs" as possible, to minimize problems in future releases. A successful test is one that uncovers undiscovered errors, not one in which troubles fail to be encountered [Ref 26]. A Developmental Test resulting in few reported problems is most likely a test that lacked rigor; the sooner problems are found and documented, the better. This is the basic credo of software testing.

Second, projects are always under strict fiscal, schedule, and functional constraints. Time and personnel are critical resources, and there is never enough of either to complete every task as thoroughly as desired. Priorities are set, and deadlines must be met.

In the case of MTWS, the Developmental Test was successful. Several problems were documented in detailed Software Trouble Reports. Most of the deficiencies discussed in this report have subsequently been tracked and corrected in subsequent developmental builds of MTWS. The purpose of this study is to suggest improved methods for verifying that system performance has indeed been improved.

Since the final test report was due within four weeks of completing the test, there was insufficient time to perform in-depth statistical analysis. Other more pressing tasks required the full attention of the development team. This study represents a continuation

of the Developmental Test analysis without such real-world constraints. Issues for discussion and recommendation raised in this report should not be construed as criticism. All observations, suggestions and critiques are offered with due sincerity since the author was primarily responsible for data collection and analysis throughout the MTWS Developmental Test.

All plots contained in this chapter were prepared using A Graphical Statistical System (AGSS). This software was provided by IBM to the Naval Postgraduate School under special licensing agreement. All supporting plots are located at the end of the section in which they are discussed.

A. CRITIQUE OF THE DEVELOPMENTAL TEST

Although much effort was expended gathering event timing data, the actual analysis was relatively limited. No statistical tests were performed to determine the confidence level of conclusions drawn from the data. Also, by dividing the event timing data into categories (i.e., on-time, two minutes late, and greater than two minutes late), ratio data was converted into less descriptive ordinal data. Thus, much of the information contained in the original data was essentially lost. More powerful comparative statistics could be employed if the data were analyzed in its original form.

The most direct measures of system performance, the IN and GC cycle lengths, were not considered in the original test plan and report. Except for the eight sample measurements that were deemed insufficient, this data was not available for thorough

examination. Analysis of IN and GC cycle lengths could provide valuable insight to the algorithmic efficiency of the simulation models.

A quantifiable performance baseline was not established for MTWS during the test. A firm baseline would reveal whether substantial progress had been realized for subsequent hardware and software configurations. Although significant performance improvements have been reported, it is difficult to state how much improvement has been made relative to the version tested during the Development Test.

Measures of performance (MOPs) which are both descriptive and readily lend themselves to statistical analysis were not defined for the Developmental Test. Such measures should be defined to support future testing. Once defined, techniques should be developed which simplify data collection and reduction for these MOPs. Although this task may necessitate design changes, steps taken to enhance the performance testability of MTWS will provide substantial benefits throughout its entire life cycle. Finally, developing a basic test scenario specifically designed to assess performance is crucial since all performance data is conditional on the test scenario. These issues will be addressed in the remaining sections of this study.

B. DATA COLLECTION AND REDUCTION

Data collection did not end with the Developmental Test. Since performance issues were highlighted as a concern, data were sought to provide additional insight to the system's timeliness and responsiveness. Retrieving the complete set of IN and GC cycle

lengths became a priority. Fortunately, all system alerts and messages generated by the MAN and MSC terminals had been saved in a file referred to as the "alert log". The UNIX "grep" command was used to extract the IN and GC data, which were intermixed with other messages. This output was spooled to a high speed printer. Again, it was necessary to search manually through a large volume of computer printouts to capture the relevant data. This required another two man-weeks of effort. A total of 169 and 1137 data readings were drawn from the alert logs for IN and GC cycle lengths respectively. This represents the entire set of both cycles during the exercise except for a few partial measurements caused by exercise suspensions and re-starts. The partial measurements were removed from the data sets.

Each GC and IN cycle length reading was time-stamped with the operating system clock time of its respective MAN workstation rather than with game-time of the exercise. However, these time values varied between the MAN 002 (Intelligence) and MAN 003 (Ground Combat) terminals since the operating systems of these workstations were not initialized simultaneously. It was necessary to convert all instances of operating system time to game-time to relate IN and GC data to the scenario. This conversion was not precise. As a result, the estimated game-time for IN and GC measurements may vary approximately one to two minutes from when the reading actually occurred.

Once event time differential, IN cycle length, and GC cycle length data had been converted to a common time scale, the data was stored in separate text files. The event time differential data was transformed to the absolute value to reflect the magnitude of the

difference between scheduled and executed times. Thus, an event occurring a minute early has the same event time differential as one occurring a minute late.

Another problem was encountered when using statistical software to construct plots of performance measures against game-time. Forty minute gaps appear on the time scale since minutes reset in increments of 60, but the software creates a scale that includes the values between 60 and 100. As a solution, all game-time date-time groups (e.g., 140600ZNOV) were converted to consecutive minutes of the Developmental Test scenario (e.g., 360 minutes elapsed time). The exercise start time of 140300ZNOV was thus assigned a value of zero minutes. UEDIT-2, a spreadsheet employing the power of the APL programming language, was used to accomplish this data conversion for all three data files [Ref. 27].

C. ASSESSMENT OF DEVELOPMENTAL TEST RESULTS

This section will assess the validity and significance of the Developmental Test findings using appropriate statistical and graphical techniques. The first part examines conclusions drawn regarding the timeliness of events. Next, the accuracy of the informal cycle length sampling method is examined in detail.

1. Event Timing

One of the most important conclusions of the test was that timing problems were most evident for ground movements [Ref. 28]. Performing a Chi-Square Test for Differences in Multinomial Probabilities on the event timing data contained in Table 3

yields a test statistic of 464.128 (eight degrees of freedom) with a p-value less than 0.0001 [Ref 29]. Thus, the hypothesis that all event types are equally likely to be on-time can be rejected with near certainty.

Graphical analysis procedures were used to identify and examine distributional differences between event types. The technique examines the components of the chi-square statistic resulting from the null hypothesis of equality of several multinomial distributions. A more thorough discussion of the theory and application of this method is provided as an appendix [Ref. 30]. In this case, there are five multinomial distributions, one for each event type, with the same three possible outcomes. Let:

i = 1 to 5 be the event type, such that

- 1 = Ground Moves (GM)
- 2 = Fire Missions (FM)
- 3 = Air Missions (AM)
- 4 = Ship Moves (SHIP)
- 5 = Ship-to-Shore Moves (STS),

j = 1 to 3 be the observation category, such that

- 1 = On-Time
- 2 = 2 Minutes Late
- 3 = Greater than 2 Minutes Late,

r = 5, the number of distributions (i.e., event types),

c = 3, the number of possible outcome (i.e., observation categories),

Y_{ij} = observed number of event type i in category j, and

m_{ij} = expected number of event type i in category j.

This yields a test statistic,

$$Q = \sum_i \sum_j \frac{(Y_{ij} - m_{ij})^2}{m_{ij}}$$

which approximates a χ^2 random variable with $(r-1)*(c-1) = 8$ degrees of freedom.

Figure 1 presents four plots which graphically portray the chi-square statistic and provide more insight to the data distributions. The plot of the relative frequency distributions (refer to Figure 1, upper, left plot) shows that the event type distributions are dissimilar. Ground moves were most likely to be greater than two minutes late while the other event types were usually on-time. The plot of observed minus expected counts (i.e., the residuals) shows that the number of late ground moves is far more than expected while the number of on-time ground moves is far less than expected. The exact opposite is true for the other event types.

When the residuals are standardized by dividing by the square root of m_{ij} , the GM residuals still dominate (refer to Figure 1, lower, right plot). This indicates that there may be major differences between ground moves and other event types. The fourth plot (Figure 1, lower left corner) depicting contributions to the chi-square statistic (the square of the standardized residuals), shows that most of the large chi-square value is due to ground moves. Thus, graphical analysis has visually confirmed that differences exist between the distributions of event types, and shown that ground moves differ significantly from the other distributions ($p\text{-value} < 0.0001$). With the high relative frequency of late GM events, timing problems were indeed most evident for ground moves.

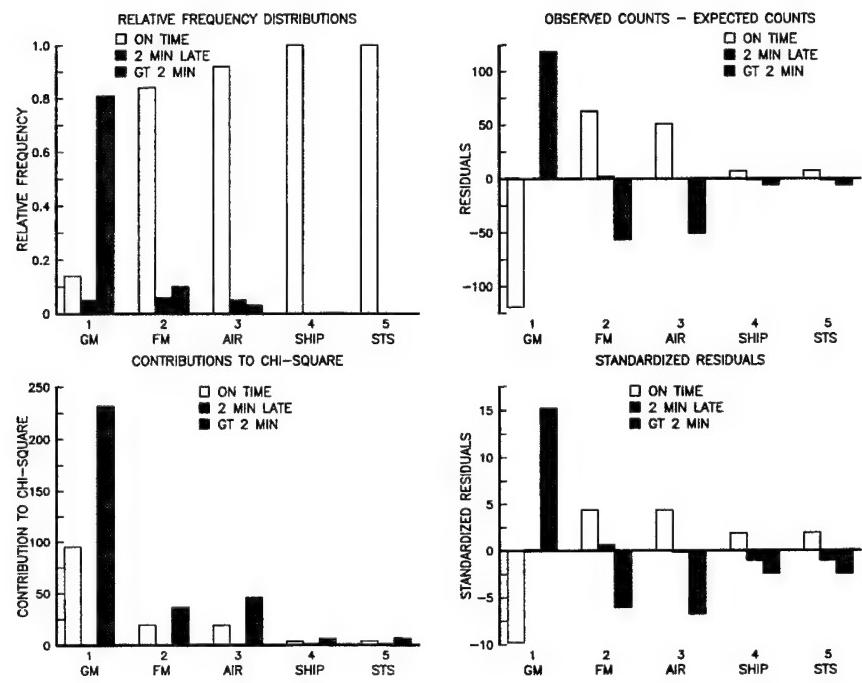


Figure 1. Chi-Square Test For Differences In Multinomial Probabilities

2. Cycle Length Sampling

Next, the question of whether the eight IN and GC cycle length samples were representative of their underlying distributions is examined. Throughout this paper, the term "population" is used to refer to all Developmental Test data actually generated using MTWS version ar115.4 and the test scenario. The "sample" refers to the eight sets of cycle length readings physically recorded during the Developmental Test. Comparing the samples to their respective population distributions will help assess the usefulness and precision of the informal sampling technique.

Figure 2 presents a comparison of the eight sample IN measurements to the total Developmental Test population of complete IN cycles using box plots. Box plots are summary displays of the data and provide an immediate look at the prominent features of the distributions (such as the mean, median, and quartile values) [Ref. 31]. Table 4 provides a comparative summary of intelligence cycle sample and population parameters for the mean, standard deviation, and quartile values. The GC sample and population distributions are similarly compared in Figure 3 and Table 5.

Considering that the samples were so small, Tables 4 and 5 show a surprising similarity between the sample and population distributions for both GC and IN cycle lengths. Many of the sample estimates for the mean, standard deviation, and quartile values do not seem to vary much from their underlying population values.

However, the apparent similarity of the samples to their respective population distributions is somewhat deceiving. Examination of the box plots reveals substantial

INTELLIGENCE CYCLE
COMPARISON OF SAMPLE TO POPULATION

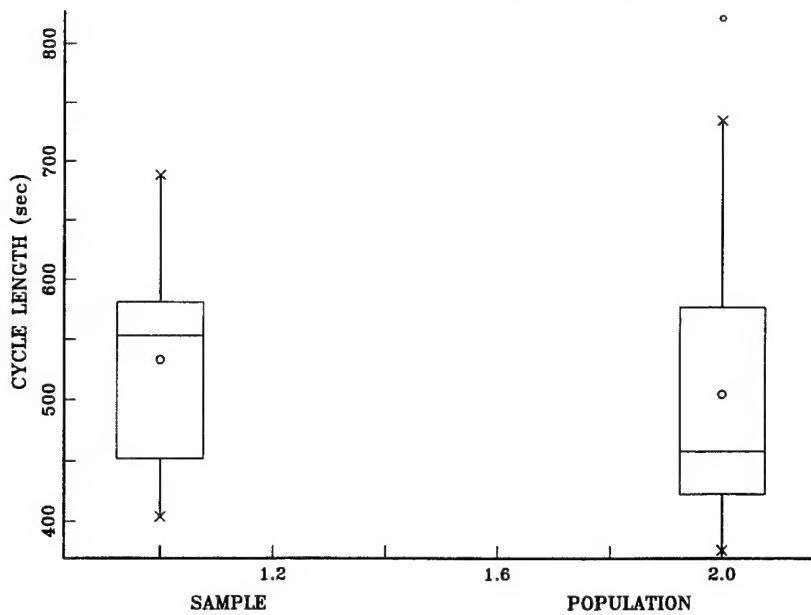


Figure 2. Intelligence Cycle Comparison Of Sample To Population

	No. Points	Mean	Std Dev	$Q_{.25}$	$Q_{.50}$	$Q_{.75}$
Sample	8	533	92.45	435	550	577
Population	169	504.84	96.91	423	458	577

**Table 4. Comparison Of Intelligence Cycle
Sample To Population**

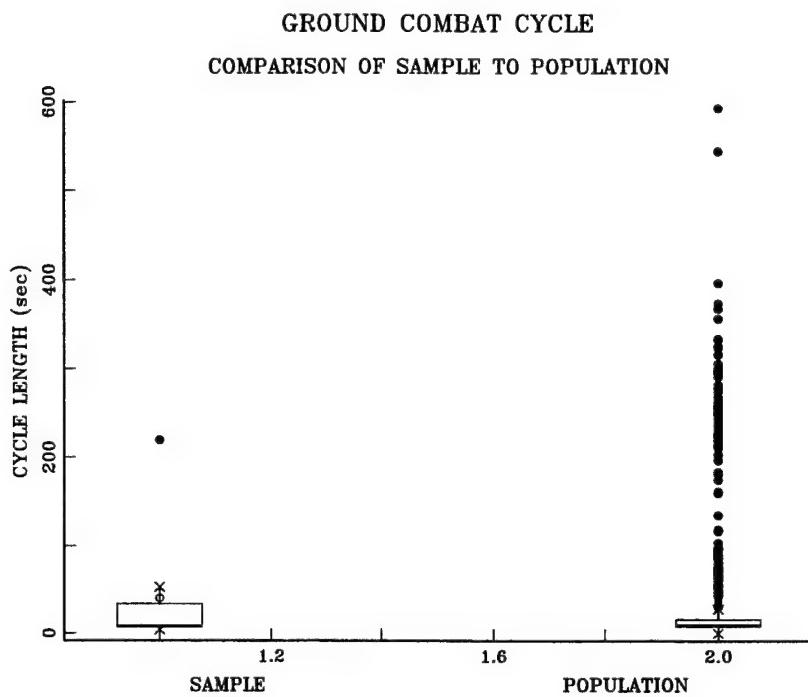


Figure 3. Ground Combat Cycle Comparison Of Sample To Population

	No. Points	Mean	Std Dev	Q _{.25}	Q _{.50}	Q _{.75}
Sample	8	40.12	74.01	6	8	14
Population	1,137	30.24	65.58	8	10	16

**Table 5. Comparison Of Ground Combat Cycle
 Sample To Population**

differences. Figure 2 shows variations within the center portion of the distributions between the IN cycle sample and its overall population. The relative position of the median within the box plots indicates that the center of the IN sample is skewed left while the center of the IN population is actually skewed to the right. The tails of the IN population also appear to be much longer than the sample. The GC data summarized in Figure 3 shows that the right tails of the sample and population distributions also differ greatly. In particular, the GC cycle population readings have several data points well beyond the upper adjacent value.

Since the box plots and accompanying tables provide only a summary of the GC and IN cycles, it is best to take a closer look at the data using empirical quantile - quantile (Q-Q) plots. Figure 4 presents a plot of the quantiles of the IN sample against the quantiles of the overall population. Figure 5A shows a similar comparison for the GC data. Since the points on Figure 5A are tightly bunched in the lower range of the scale, a \log_{10} transformation was performed on the GC data to expand the plot over a the range of values. The Q-Q plot of the transformed GC data is displayed in Figure 5B.

Figure 4 shows that the quantiles of the IN sample do not seem to lie near those of its parent distribution. Deviations in the center portion of the range of values are particularly evident. For the GC data, Figures 5A and 5B show that the eight sample measurements are fairly representative of the lower quantiles of the overall distribution. However, significant departures from the $y = x$ line, which represents equality between the quantiles of the compared distributions, seem to occur in the upper range of values.

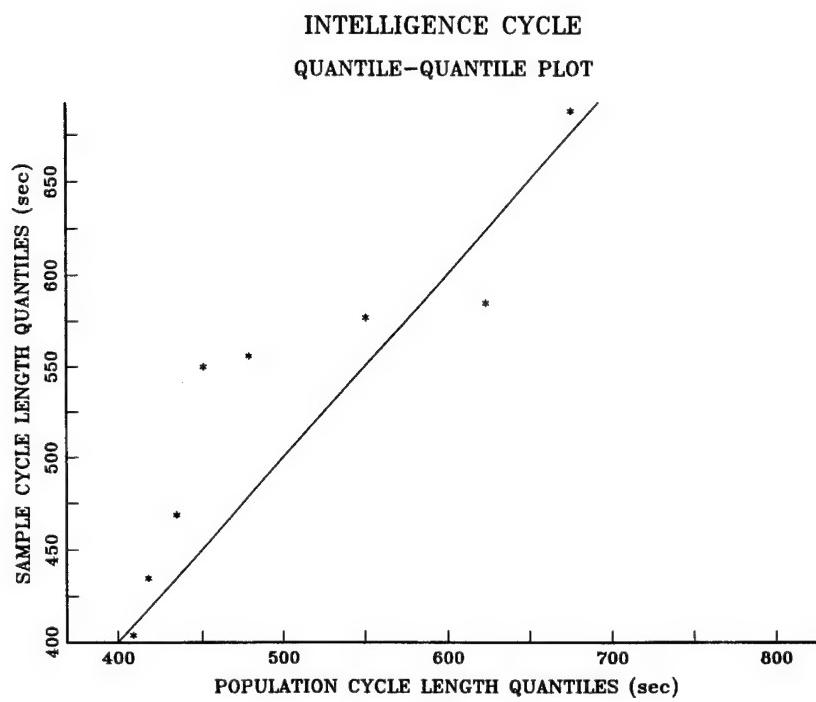


Figure 4. Intelligence Cycle Quantile-Quantile Plot

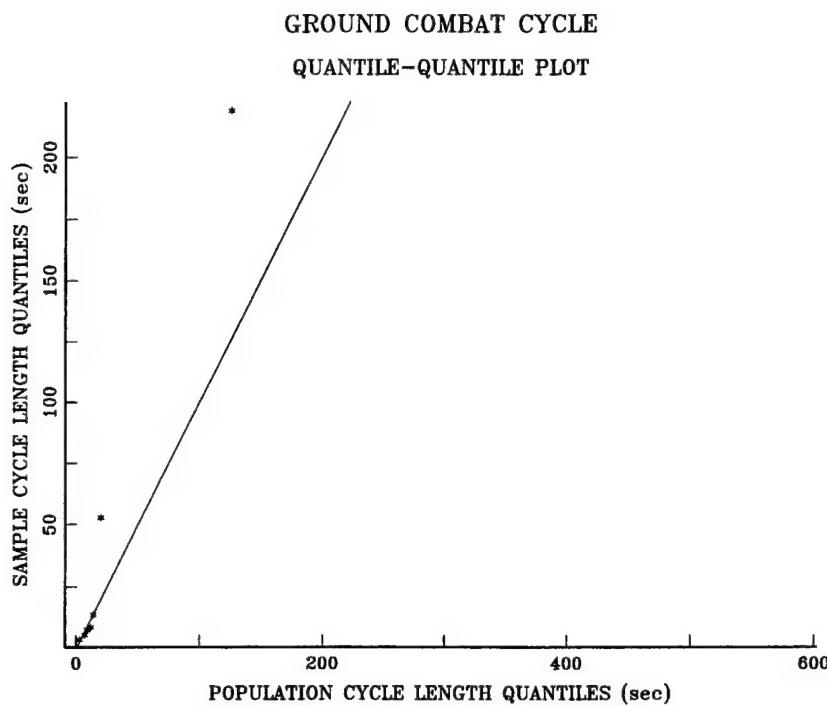


Figure 5A. Ground Combat Cycle Quantile-Quantile Plot

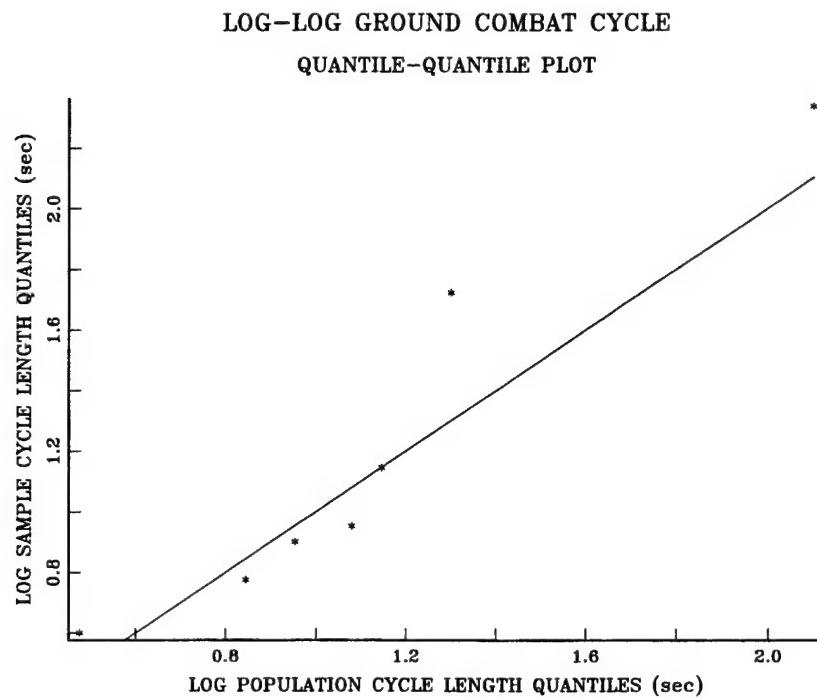


Figure 5B. Log-Log Ground Combat Cycle Quantile-Quantile Plot

Thus, the Q-Q plots further evidence the departures of the samples from their respective population distributions previously noted in the box plots.

Unlike standard statistical tests, the empirical Q-Q plots reveal differences over the entire distribution rather than just the center quartiles. These plots indicate that the informal sampling technique employed during the Developmental Test did not provide truly representative data. The three hour sampling technique may be adequate for cursory analysis, but eight samples will not yield reliable population estimates as a rule. It was wise not to draw findings based on such a limited sample. As a result of this analysis, the need to develop improved data collection methods for IN and GC cycle lengths becomes apparent.

D. DETAILED ANALYSIS

In this section, graphical analysis of the Developmental Test data will be used to identify relationships regarding system performance. This should provide insights toward developing valid measures of performance and appropriate testing techniques. Three main areas of performance are examined: 1) the IN Cycle Length; 2) the GC Cycle Length; and 3) the Event Time Differential. The observations noted in this section will be further explored in Section E; this is where operational causes affecting performance will be discussed in terms of the scenario.

1. Intelligence Cycle

Figure 6 presents a scatter plot of the IN cycle data with a LOWESS curve. LOWESS, which stands for locally weighted regression scatter plot smoothing, provides an accurate impression of dependence of the Y on X variables over the range of data [Ref. 32]. Table 6 provides a statistical summary of this data. The maximum cycle length, occurring at 1148 minutes (H+5:08 in the scenario), was 821 seconds. The minimum cycle length of 376 seconds occurred at 1134 minutes (H+4:54). To visualize how the IN cycle varied over time, a strip box plot was prepared with the data segmented into two-hour bins; this plot is presented in Figure 7. Periods of high cycle length indicate when most stress was placed on the intelligence algorithm by the test scenario. Table 7 identifies the maximum values and the period in which they occurred for selected descriptive statistics for the data from the individual box plots.

INTELLIGENCE CYCLE
SCATTER PLOT OF CYCLE LENGTH vs. TIME

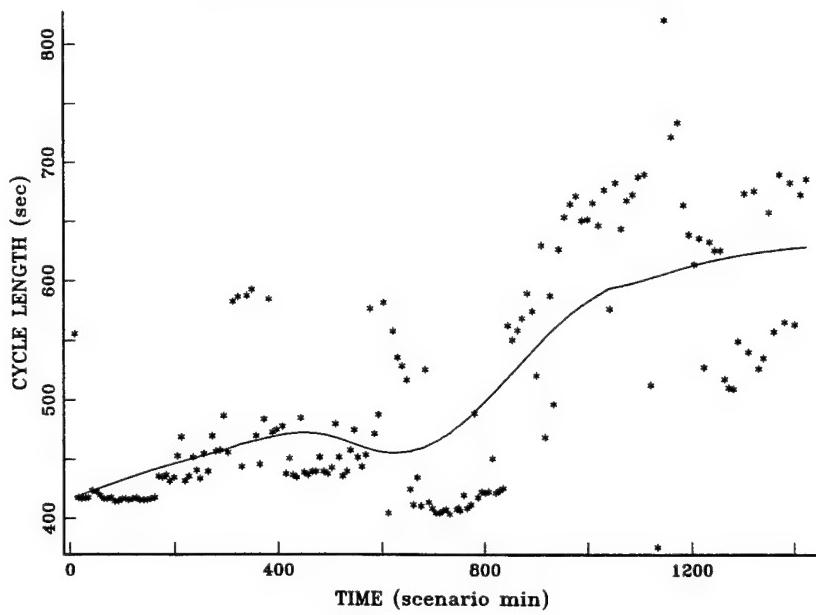


Figure 6. Intelligence Cycle Scatter Plot

No. Points	Mean	Std Dev	$Q_{.25}$	$Q_{.50}$	$Q_{.75}$
169	504.84	96.91	423	458	577

**Table 6. Intelligence Cycle
Data Summary**

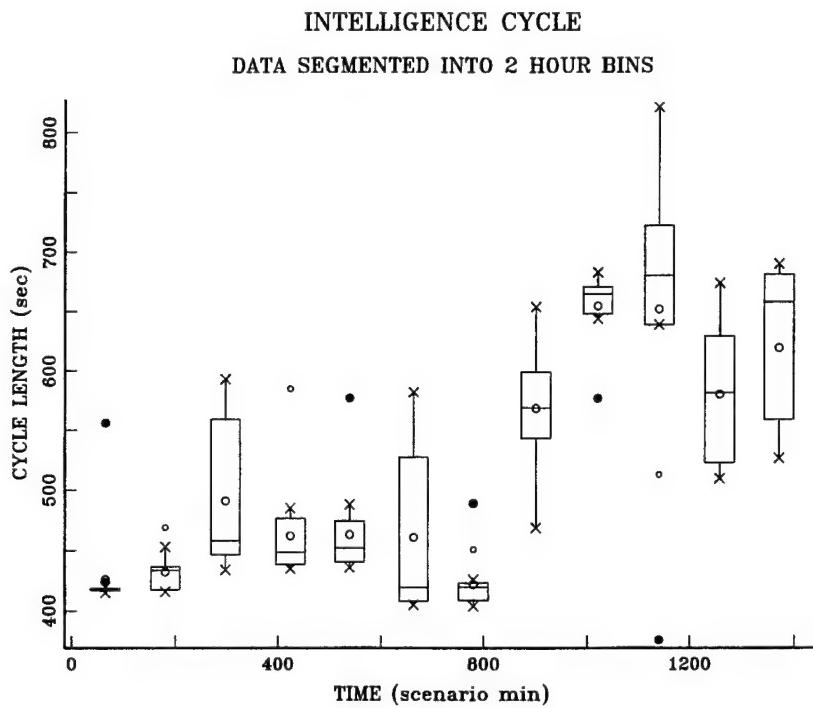


Figure 7. Intelligence Cycle Strip Box Plots

Stress Measure	Maximum Value	Period (minutes)	Period (H-hour)
Mean	655	967 to 1075	H+2:07 to H+3:55
Median	673	1086 to 1195	H+4:06 to H+5:55
Std Deviation	124	1086 to 1195	H+4:06 to H+5:55

**Table 7. Intelligence Cycle
Periods Of Highest Stress On Performance**

During these periods of high activity, the system required approximately 11 minutes to determine which game objects could detect each other. To see how this may affect the play of the game, an operational context is necessary. The worst case ground scenario would be a movement to contact between opposing mechanized forces in flat and open terrain, such as a desert. Assuming both forces advance in tactical formation toward one another at the rate of 25 kilometers per hour (kph), the closure rate between forces would be 50 kph. In the 11 minutes required to complete one intelligence cycle under peak load, the forces would cover a combined distance of more than nine kilometers. Thus, the converging forces could conceivably pass without firing a shot or detecting one another even though they started out well beyond direct fire engagement range. This serious deficiency has since been remedied, but was a matter of great concern during the Development Test. Although test participants considered the IN cycle lengths to be excessive, the true extent of the problem could not be accurately assessed then due to the lack of data. It is now apparent that the demands placed by the scenario on the intelligence algorithm were greatest from H+2 to H+6 and resulted in degraded performance of the combat simulation.

2. Ground Combat Cycle

A scatter plot of the GC cycle data with a LOWESS curve is shown in Figure 8.

Table 8 provides a statistical summary of this data. The maximum cycle length of 593 seconds occurred at 1150 minutes (H+5:10 in the scenario). The minimum cycle length of two seconds occurred two minutes into the exercise. A strip box plot showing GC data divided into two-hour time segments is presented in Figure 9. Once again, periods of high cycle length show when most stress was placed on the system by the test scenario. Table 9 lists the maximum values and time of occurrence for selected descriptive statistics drawn from the individual box plots.

During the high stress periods, the system required approximately 4.5 minutes to determine the results of ground engagements and to assess casualties. Using the worst case scenario previously discussed, a vehicle that should have received a catastrophic hit could possibly advance another 1.87 km toward the enemy, continuing the battle before being destroyed. This flaw represents a serious departure from reality which can undermine the validity of the simulation. Fortunately, this problem has also been corrected in more recent MTWS versions. However, the test scenario placed most stress on the ground combat algorithm from H+4 to H+8 (1080 to 1320 minutes); this is when the performance of version ar115.4 lagged significantly during the Developmental Test.

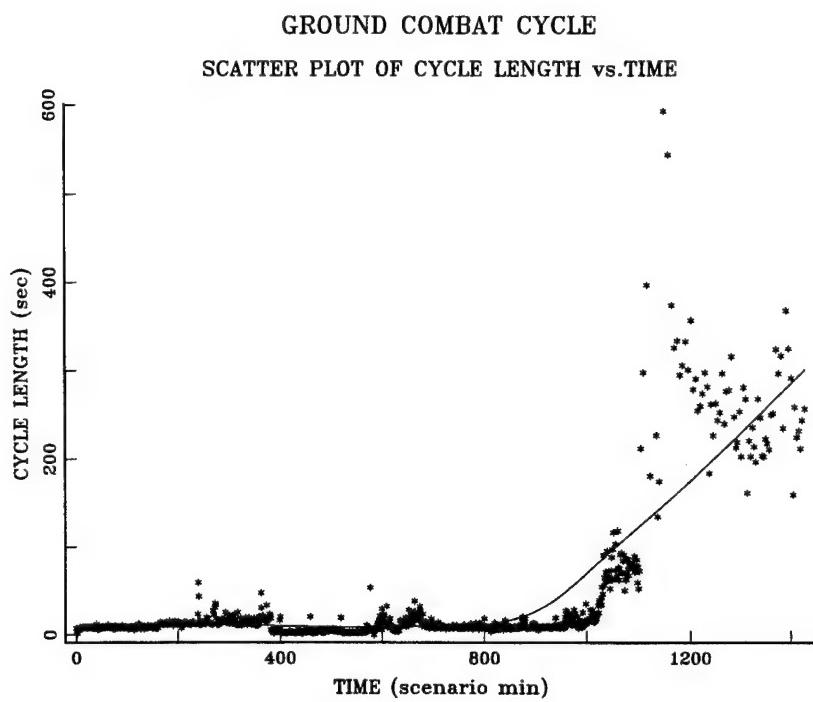


Figure 8. Ground Combat Cycle Scatter Plot

No. Points	Mean	Std Dev	$Q_{.25}$	$Q_{.50}$	$Q_{.75}$
1,137	30.24	65.58	8	10	16

**Table 8. Ground Combat Cycle
Data Summary**

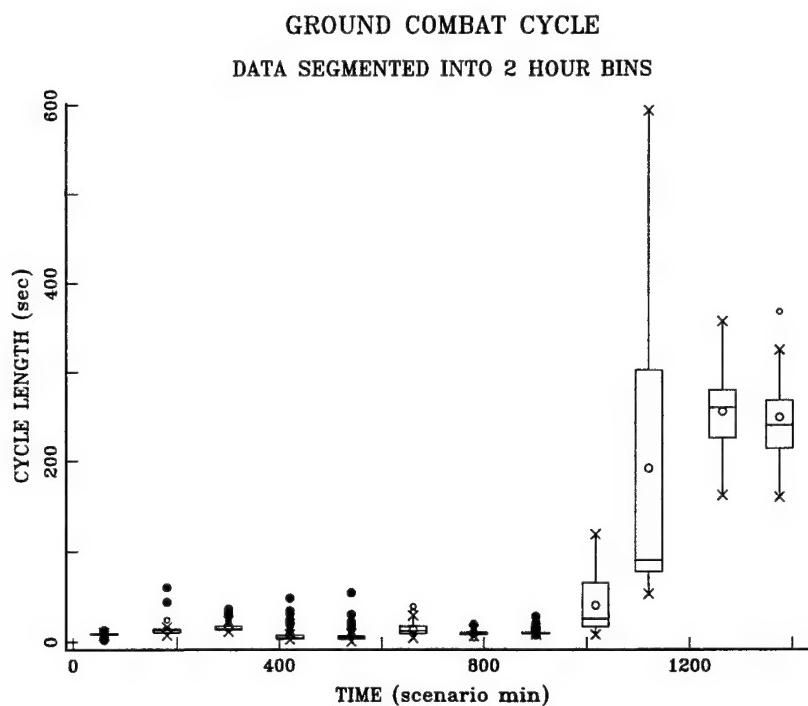


Figure 9. Ground Combat Cycle Strip Box Plots

Stress Measure	Maximum Value	Period (minutes)	Period (H-hour)
Mean	255	1203 to 1320	H+6:03 to H+8:00
Median	260	1203 to 1320	H+6:03 to H+8:00
Std Deviation	124	1081 to 1197	H+4:01 to H+5:57

**Table 9. Ground Combat Cycle
Periods Of Highest Stress On Performance**

3. Event Time Differential

Figure 10 is a scatter plot of the event time differential data with a LOWESS curve. It was necessary to randomly jitter the data values by 1.5% in each dimension to reduce over-plotting. Table 10 provides an aggregate summary of this data for all Developmental Test events. The maximum delay for a scheduled event was 22 minutes occurring at 1180 minutes (H+5:40 in the scenario). The minimum timing differential of zero occurred frequently throughout the exercise. In fact, zero was the mode of the event timing distribution. A strip box plot of this data segmented into two hour bins is displayed in Figure 11. The plots reveal that most timing problems occurred roughly between 1000 and 1200 minutes (H+2:40 to H+6:00) into the scenario. The largest mean, median, and standard deviation of the two hour blocks were all observed in the 970 to 1080 minute period (H+2:10 to H+4:00) as highlighted in Table 11. These delays were the result of the scenario placing high computational demands on the system.

However, the analysis of event timing data is not a simple matter. Section C of this chapter demonstrated that the event time differential is dependent on the type of event, and that ground moves were most likely to be delayed. Figure 12 provides a summary of event time differential data with respect to the five event types: ground moves, fire missions, air missions, ship moves, and ship-to-shore moves. Table 12 highlights the differences between event types and provides an interesting contrast to the original Developmental Test results presented in Table 3.

Whereas Table 3 presented timing data as ordinal counts, Table 12 provides descriptive statistics based on a ratio measurement scale. The distribution of ground moves is clearly different from the distributions of the other events. Ground moves had the highest mean and median time differential values, 9.46 and 5 respectively. All other events had a mean value less than 1 minute and a median value of 0 minutes.

Although Figure 12 and Table 12 both show how event time differential data varies according to event type, it does not reveal the interaction of events with respect to time. This requires a view with an additional dimension. Figure 13 provides a three-dimensional perspective of how the events unfolded during the Development Test.

Points representing separate events are plotted according to their scheduled time, type of event, and event time differential (i.e., scheduled time - execution time). The greatest timing delays occurred when multiple ground units commenced simultaneous movement at 1080 minutes (H+4). Fire missions seem more likely to be late when scheduled concurrent with ground movement. The effect of ground movements on air missions is noticeable but less pronounced. This is somewhat contrary to the results of the original test report. It seems that the occurrence of late air missions was more evenly distributed over time. This is probably because the scheduling of air missions was less closely linked to ground movement in the scenario. The tactical scenario merely reflects current Marine Corps doctrine on this point. Fire missions are usually scheduled to support ground maneuver while air strikes are employed continuously to shape the battlefield over the entire area of interest.

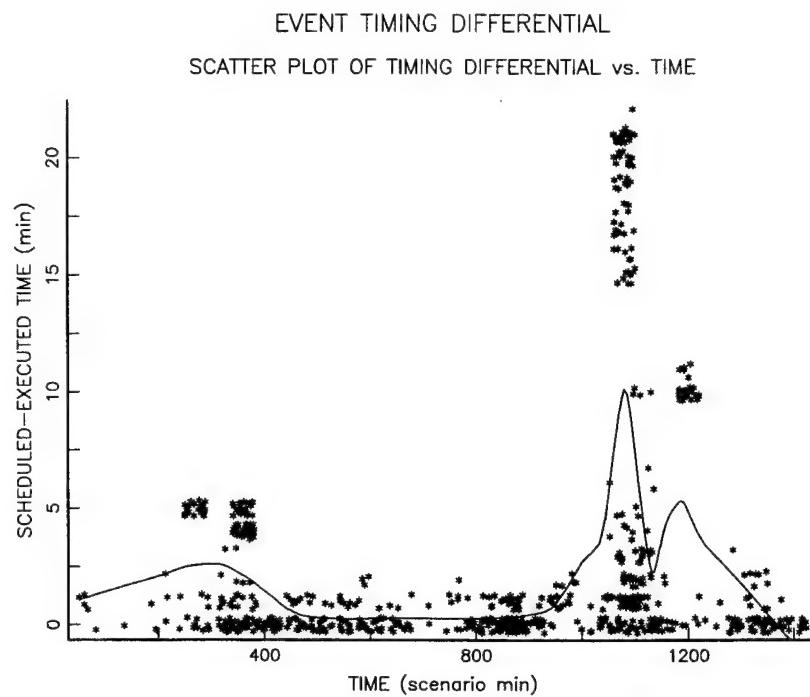


Figure 10. Event Timing Differential Scatter Plot

No. Points	Mean	Std Dev	$Q_{.25}$	$Q_{.50}$	$Q_{.75}$
781	3.11	5.73	0	1	4

**Table 10. Event Time Differential
Data Summary**

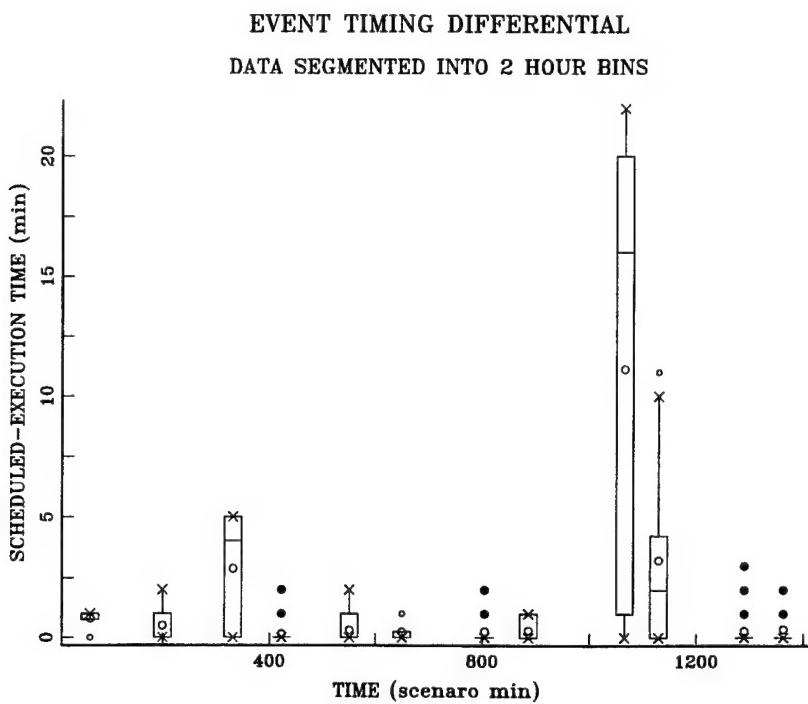


Figure 11. Event Timing Differential Strip Box Plots

Stress Measure	Maximum Value	Period (minutes)	Period (H-hour)
Mean	11.13	970 to 1080	H+2:10 to H+4:00
Median	16	970 to 1080	H+2:10 to H+4:00
Std Deviation	9.18	970 to 1080	H+2:10 to H+4:00

**Table 11. Event Time Differential
Periods Of Highest Stress On Performance**

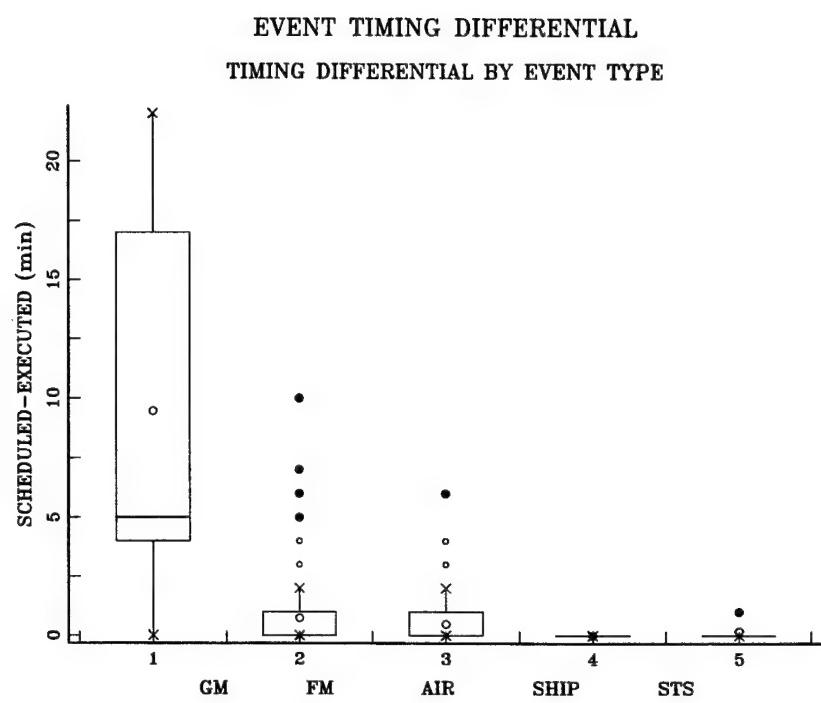


Figure 12 Event Timing Differential Box Plots By Event Type

Event Type	No. Events	Mean	Std Dev	Q_{.25}	Q_{.50}	Q_{.75}
Air Missions	201	0.48	0.81	0	0	1
Fire Missions	313	0.73	1.56	0	0	1
Ground Moves	222	9.46	7.42	4	5	17
Ship-to-Shore	23	0.17	0.39	0	0	0
Ship Moves	22	0	0	0	0	0
All Events	781	3.11	5.73	0	1	4

**Table 12. Event Time Differential
Data Summary By Event Type**

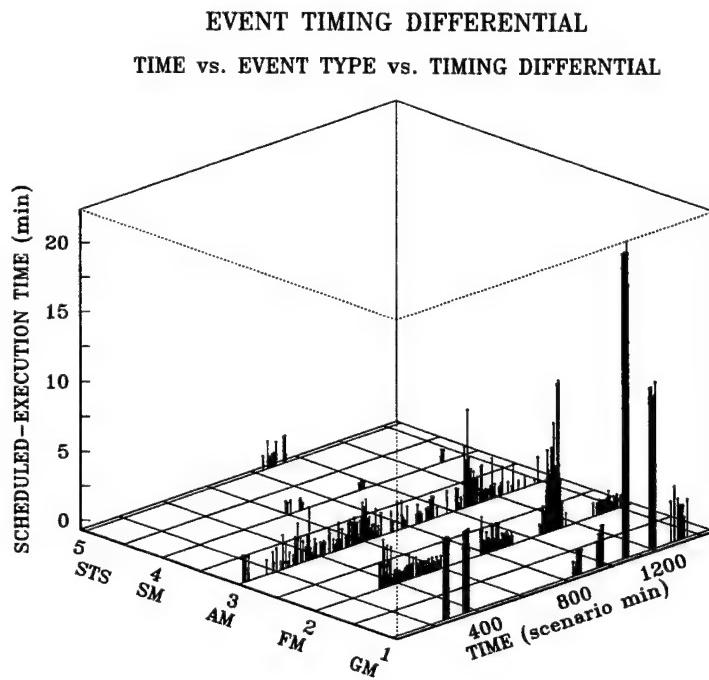


Figure 13. Event Timing Differential 3-D Plot Of Scenario

E. DISCUSSION

The best time to assess performance of a simulation is during periods of high stress. This is when performance is likely to lag, if at all. The ability to reduce timing troubles and handle heavy stress is what distinguishes one version as performing better than another. However, the computational load placed on the system is governed by event scheduling in the scenario. A common scenario must be employed to make valid comparisons between different system configurations, e.g. to quantify the effects of hardware upgrades or software changes. This highlights the need to identify and baseline a scenario that is sufficiently rigorous to conduct performance testing of MTWS over the system's life cycle.

Although the Developmental Test included more than 24 hours of tactical play, the scenario placed high stress on MTWS for only a portion of that time. A review of the plots in the preceding section shows that the level of activity was relatively constant for the first 900 minutes of the exercise. This holds true for all performance measures to include the IN cycle length, GC cycle length, and the event time differential. The six hour period from 960 to 1320 minutes placed the maximum demand on the system. The task of performance testing could be greatly simplified by running the Developmental Test scenario only from H+2 to H+8. The other portions of the scenario, while useful for studying capacity, reliability, and functionality, provide little insight to the performance of MTWS. If performance is the primary testing concern, there is little need to tie up critical

equipment and personnel for more prolonged periods. Additional tests can be designed to focus on other aspects of the system specification.

A detailed examination of the scenario yields many interesting insights. Table 13 highlights some of the significant events in the exercise. Times are listed in terms of scenario minutes, H-hour, local game-time (I time zone), and Zulu game-time. A night time amphibious assault was conducted by a regimental landing team at H-hour which was set at 0200 local time. This was followed by a pre-dawn main attack at 0600. Since darkness prevailed over the battlefield, detections were limited even though opposing forces were in close proximity as the MEF advance continued. This situation began to change as the model simulated the transition from night to day. Beginning morning nautical twilight (BMNT) occurred at 0648 local time which signaled the start of a strong "Dawn Effect" in the simulation that continued through sunrise.

The gradual rise of the sun placed extreme stress on the intelligence algorithm as visual detection ranges rapidly increased with the passing of darkness. The IN algorithm was forced to perform numerous detection updates, with greater ranges for line of sight calculations, for almost all units and collection assets over a relatively brief period of time. This resulted in a maximum IN cycle length of more than 821 seconds at 0708 local time, mid-way between BMNT and sunrise. When the number of ground detections between the opposing forces increased, a heavy demand was then placed on the ground combat algorithm. The GC cycle quickly jumped to a maximum value of 593 seconds at 0710 as

Time in Scenario Min	Time in H-hour	Event	Local Game Time	Zulu Game Time
0	H-14:00	Start Exercise	141200INOV	140300ZNOV
270	H-9:30	CSS Operations	141630INOV	140730ZNOV
360	H-8:00	Ground Advance	141800INOV	140900ZNOV
840	H+0:00	H-hour	150200INOV	141700ZNOV
1,050	H+3:30	NGF & Air Msn	150530INOV	142030ZNOV
1,080	H+4:00	Main Attack	150600INOV	142100ZNOV
1,128	H+4:48	BMNT	150648INOV	142148ZNOV
1,168	H+5:28	Sunrise	150728INOV	142228ZNOV
1,200	H+6:00	CSS Operations	150800INOV	142300ZNOV
1,290	H+7:30	NPKA Attacks	150930INOV	150030ZNOV

Table 13. Key Scenario Events Affecting Performance

ground engagements erupted across the MTWS battlefield. The combination of the massive ground attacks followed closely by sunrise was responsible for the high computational demands noted during this period. These events demonstrated the performance limitations of the overall simulation while highlighting the need for improvement. These problems may not have surfaced with a less demanding scenario. Fortunately, the performance deficiencies noted with the early version tested at the Developmental Test have been corrected in subsequent MTWS builds. However, as the simulation grows in size and complexity, problems with timeliness and responsiveness of the system are likely to re-occur.

V. RECOMMENDATIONS

A. MEASURES OF PERFORMANCE

Measures of performance (MOPs) for MTWS should adequately summarize key aspects of the simulation's responsiveness. Ideally, these standards should be stipulated in the requirements or specification documents. However, MTWS documentation emphasizes functional and capability requirements rather than performance. Thus, MOPs are still evolving as the system progresses from development to fielding. This section will present specific recommendations that can be used to assess MTWS performance over its life cycle.

MOPs should reflect both statistical and operational considerations. Measures should facilitate the statistical comparison between various MTWS configurations to determine if the performance of one is indeed significantly different from another. Since the most powerful statistical tests are based on data with interval or higher measurement scale, it is highly desirable to develop measures based on that level of precision. Statistics such as the mean and standard deviation are usually sufficient for conducting such analysis. However, quantifiable operational standards should also be established to distinguish acceptable from unacceptable performance. These thresholds should have real-world relevance and serve as benchmarks for assessing the fidelity of the system. The percentages of events that exceed these threshold values also become important test statistics.

Table 14 is a listing of proposed performance standards for MTWS based on various system processes and scenario events. The threshold value for the IN cycle is based on the movement to contact scenario presented in the previous chapter. Recall that two hostile units are rapidly advancing across desert-like terrain. In this situation, the system should complete detection updates before the converging forces traverse 1000 meters, which is the range of most medium machine-guns and medium anti-armor missiles. At this point

Activity/Event	Performance Standard
1. Detection Updates (IN cycle)	Cycle time not more than 72 seconds
2. Ground Combat Updates (GC cycle)	Cycle time not more than 50 seconds
3. Ground Moves	Time differential not more than 1 minute
4. Fire Missions	Time differential not more than 1 minute
5. Air Missions	Time differential not more than 1 minute
6. Ship Moves	Time differential not more than 1 minute
7. Ship-to-Shore Moves	Time differential not more than 1 minute

Table 14. Recommended Performance Standards

the nature of the battle and the weapons that can be brought to bear change significantly. With a closure rate of 50 kph between opposing mechanized forces, 1000 meters can be covered in 72 seconds. As a result, the play of the game will be degraded whenever the IN cycle length exceeds 72 seconds in this worst case situation.

Similarly, ground combat updates should be completed before a moving vehicle can pass through effective small arms and light anti-armor missile range of approximately 350 meters. Since these weapons are appropriate only for dismounted forces, one force must be relatively stationary, so the closure rate drops to 25 kph. At this rate of advance, 350 meters will be traversed in 50 seconds. In this case, performance of the simulation lags when the GC cycle exceeds 50 seconds.

Although both proposed standards are admittedly subjective, they represent an attempt to relate run-time computational measures to actual combat capabilities. It is necessary to specify a demanding tactical situation and make reasonable simplifying assumptions if these thresholds are to have real meaning. Processing delays become significant when the tactical nature or results of the battle are substantially altered; this is best defined in terms of weapon capabilities. And, since the range and lethality of weapon systems have invariably increased over time, these values should be periodically re-evaluated during the life cycle of MTWS. What makes tactical sense today may soon become outdated by technological advances.

Items three through seven in Table 14 echo the standards first defined in the Developmental Test Plan. A tolerance of one minute makes sense in operational terms since these events are not executed precisely on-time in real combat. For example, the odds are minuscule that all units in a division-sized attack will cross the line of departure exactly at H-hour. Likewise, it is rare to execute an air mission or fire mission within a few seconds of the desired time on target. Variance of timing is a part of real battle, although

there are few studies which address this issue based on discussions with members of the combat modeling community. Since MTWS spot reports are rounded to the nearest minute, the logical choice lies between a threshold value of zero or one. A one minute tolerance seems reasonable for the vast majority of events. Having proposed this general rule for assessing event timeliness, there are definitely circumstances for which a delay up to one minute would be unacceptable. However, a stronger case can be made that zero tolerance for delays would be completely unrealistic in many more situations.

Proposed measures of performance for assessing MTWS are listed in Table 15. These include summary statistics as well as measures derived from performance standards. The mean and standard deviation provide the most precise measures for the location and variability of symmetric, or nearly symmetric, data distributions. As such, these statistics are the basis of standard parametric statistical procedures. However, if the data distributions appear to be asymmetric or to have more than one mode, the median value should be used as a measure of central tendency rather than the mean, and the inter-quartile range (IQR) should be used as a measure of spread rather than the standard deviation.

Since the detailed analysis section highlighted the difference between the event types, test statistics should be maintained separately for each event type rather than aggregated. This will provide a more accurate view concerning the timeliness and responsiveness of the system. The performance MOPs listed in Table 15 are descriptive and quantifiable, and will support a variety of powerful statistical tests. Together, they will provide a

Measure of Performance	Statistical/Threshold Measures
1. Detection Updates (IN cycle)	a. Mean Cycle Length b. Median Cycle Length c. Standard Deviation of Cycle Length d. IQR of Cycle Length e. Proportion of Cycle Lengths > 72 sec
2. Ground Combat Updates (GC cycle)	a. Mean Cycle Length b. Median Cycle Length c. Standard Deviation of Cycle Length d. IQR of Cycle Length e. Proportion of Cycle Lengths > 50 sec
3. Ground Moves	a. Mean Time Differential b. Median Time Differential c. Standard Deviation of Time Differential d. IQR of Time Differential e. Proportion of Time Differential > 1 min
4. Fire Missions	a. Mean Time Differential b. Median Time Differential c. Standard Deviation of Time Differential d. IQR of Time Differential e. Proportion of Time Differential > 1 min
5. Air Missions	a. Mean Time Differential b. Median Time Differential c. Standard Deviation of Time Differential d. IQR of Time Differential e. Proportion of Time Differential > 1 min
6. Ship Moves	a. Mean Time Differential b. Median Time Differential c. Standard Deviation of Time Differential d. IQR of Time Differential e. Proportion of Time Differential > 1 min
7. Ship-to-Shore Moves	a. Mean Time Differential b. Median Time Differential c. Standard Deviation of Time Differential d. IQR of Time Differential e. Proportion of Time Differential > 1 min

Table 15. Recommended Measures Of Performance

summary of sufficient detail to make valid comparisons between various configurations of MTWS.

B. DATA COLLECTION

Previous chapters have discussed the collection of data to include event time differential, IN cycle length, and GC cycle length. In total, this task required four man-weeks of effort for this study. To enable timely analysis, the time spent collecting data must be reduced to a matter of days or even hours. This can be achieved through automating the data collection effort.

Instead of employing manual collection techniques, relevant data could have been written to separate output files. Table 16 lists the pertinent performance data that should be stored in each file. These data elements are required to calculate the MOPs defined in the preceding section.

Data File	Data Elements
Event Time Differential	All Spot Reports
Intelligence Cycle	Cycle Number Cycle Length Game-Time (Zulu)
Ground Combat Cycle	Cycle Number Cycle Length Game-Time (Zulu)

Table 16. Performance Data Files

Once stored in electronic form, the data can be searched or manipulated as necessary using a variety of tools. Data collection and reduction could easily be completed in four days rather than four weeks. Such methods would not only save time, but would improve the accuracy of the data as well. The possibility of transcription or time conversion errors would be substantially diminished. Automated data collection procedures limit the chance of introducing human errors into the data.

Since the Development Test, MTWS has added the capability to save spot reports to file. This will greatly reduce the time needed to gather event timing differential data in the future. However, the ability to selectively collect IN and GC cycle length data should be added as well. This would require little programming effort when compared to the benefits that would result over the long run. As the size, scope, and complexity of the system increase, performance testing will continue to be a crucial part of the MTWS program throughout its life cycle. Improved collection techniques will vastly facilitate the ability to assess the timeliness and responsiveness of the simulation processes.

C. PERFORMANCE TESTING SCENARIO

All MOPs are dependent on the scheduling of scenario events. The test scenario dictates the conditions under which the MOPs are determined. A comparison of MOPs drawn under different test conditions would be of dubious value. Therefore, it is essential to develop a standard performance testing scenario to serve as a common baseline for such analyses.

To identify an ideal performance testing scenario, many attributes should be considered. First, the test duration would need to be sufficient to produce a significant number of sample measurements without requiring critical resources for excessive durations. The scenario should place high computational demands on the system so that performance under near peak loads may be properly evaluated. An appropriate number of relevant database objects and tactical events must be included in the play of the game. Finally, the test conditions should be tightly controlled to ensure the results of the testing are repeatable.

Considering these criteria, batch files should be used exclusively to execute the test scenario. Batch files provide the control necessary to achieve reproducible test runs. They also enable the near simultaneous entry of multiple commands that can create the heavy computational loads needed to assess performance. There should be no allowance for entering "free-play" commands by operators. Permitting operators the latitude to input commands could introduce an unnecessary source of variation in the performance data.

The MTWS project should consider using the hours of H+2 to H+8 (minutes 960 through 1320) of the Developmental Test scenario as the basis for MTWS performance testing. This recommendation offers several immediate advantages. Chapter IV pointed out that this was the period when maximum demand was placed on the system. Sufficient information can be gathered in this six hour period to draw viable inferences regarding the timeliness of the most crucial events and processes. By decreasing the exercise from 24 to six hours, the data collection and reduction effort would be further reduced as an added

benefit. In this way, performance testing can be completed in less than one day rather than three days.

Many compelling reasons exist to adopt this proposal. The Developmental Test scenario is readily available and familiar to most personnel associated with the MTWS program. The batch files have already been prepared and would not require modification. However, it would be necessary to update the initial exercise database to reflect the tactical situation at H+2. This can easily be accomplished by using the MTWS "Database Save" capability to capture the state of the system at H+2 [Ref. 33]. Most importantly, sufficient data now exists to establish a quantifiable baseline for this test scenario; this will be accomplished in the next section.

This is not to suggest that the six hour segment of the Developmental Test scenario is a panacea for MTWS performance testing. It could be enhanced with a few improvements over time. First, this part of the scenario lacks an amphibious operation. Since amphibious operations are an essential part of expeditionary warfare, it will be important to determine the effects of future MTWS configuration changes on ship-to-shore movements, just as it is for other event types. It may be possible simply to re-schedule the existing Regimental Landing Team amphibious assault to occur a few hours later; the effects of such a change should be examined more closely. The number of ship moves should also be increased to provide more data on the timeliness of this event type. This task could be accomplished in conjunction with adding the amphibious assault. Finally, the test scenario could be improved by adding several database objects believed to affect the timeliness of ground

movements. This includes objects such as obstacles, barriers, unit boundaries, and fire support coordination measures [Ref. 34]. The quantities of these objects should be varied over the course of performance testing. With proper planning and control, these changes can be effected in an experiment to assess their impact on system performance.

In the meantime, hours H+2 to H+8 of the Developmental Test scenario can be a valuable tool for charting MTWS performance gains relative to the version tested in November 1994. This segment provides a rigorous test environment for evaluating the run-time efficiency of the intelligence algorithm, ground combat algorithm, and ground movement. These aspects are now the primary areas of concern. As improvements to the existing scenario are made, an updated version can be baselined to meet future performance testing requirements.

D. PERFORMANCE BASELINE

Section C of this chapter recommended using the Developmental Test scenario from H+2 to H+8 for performance testing. Section B had previously proposed specific MOPs for MTWS performance. Table 17 presents a performance baseline founded on these recommendations for the MTWS version ar115.4. MOPs for ship-to-shore movements were not computed since events of this type were not scheduled from H+2 to H+8. It is also difficult to assess the performance of ship moves with only five observations; adding more ship moves to the scenario would definitely improve the precision of these MOPs.

Measure of Performance	No. of Points	Statistical/Threshold Measures	Baseline Value
1. Detection Updates (IN cycle)	33	a. Mean Cycle Length (sec) b. Median Cycle Length (sec) c. Standard Deviation of Cycle Length (sec) d. IQR of Cycle Length (sec) e. Percent of Cycle Lengths > 72 sec	626.94 647.00 84.18 96.00 100%
2. Ground Combat Updates (GC cycle)	171	a. Mean Cycle Length (sec) b. Median Cycle Length (sec) c. Standard Deviation of Cycle Length (sec) d. IQR of Cycle Length (sec) e. Percent of Cycle Lengths > 50 sec	106.01 67.00 113.81 154.00 59.1%
3. Ground Moves	117	a. Mean Time Differential (min) b. Median Time Differential (min) c. Std Deviation of Time Differential (min) d. IQR of Time Differential (min) e. Percent of Time Differential > 1 min	14.67 17.00 6.67 10.00 91.4%
4. Fire Missions	118	a. Mean Time Differential (min) b. Median Time Differential (min) c. Std Deviation of Time Differential (min) d. IQR of Time Differential (min) e. Percent of Time Differential > 1 min	1.66 1.00 2.18 2.00 7.3%
5. Air Missions	65	a. Mean Time Differential (min) b. Median Time Differential (min) c. Std Deviation of Time Differential (min) d. IQR of Time Differential (min) e. Percent of Time Differential > 1 min	0.57 0.00 0.12 1.00 13.8%
6. Ship Moves	5	a. Mean Time Differential (min) b. Median Time Differential (min) c. Std Deviation of Time Differential (min) d. IQR of Time Differential (min) e. Percent of Time Differential > 1 min	0.00 0.00 0.00 0.00 0.0%

Table 17. Performance Baseline For Version 115.4

However, this baseline clearly reveals the severe strain placed on the intelligence and ground combat algorithms. In particular, the mean and standard deviation of the ground combat cycles were significantly higher than other portions of the scenario. The measures based on operational standards highlight where performance gains must be realized. When compared to the data in Table 3, the percentages of late air missions and fire missions have roughly doubled. The difficulties with ground movements are also more evident. In summary, these MOPs provide better insight to the timeliness of events and the responsiveness of the simulation processes.

If the same segment of the test scenario is run, these figures can be compared with current releases of MTWS to assess performance improvements. This data may support a variety of standard statistical tests such as the t-test or comparable nonparametric tests. If a more detailed comparison is desired, the data can be graphically compared by quantile-quantile plots. The point is that the performance of a new version can be quantifiably assessed using this information as a basis for comparison.

E. LESSONS LEARNED

This section will address lessons learned as they apply to MTWS as well as to combat simulations in general. Specific issues relating to system performance will be discussed in terms of the specification, design, and testing of computer-based warfare simulations.

1. Specification

Desired characteristics of system performance should be stipulated in the program requirements or system specification documents. Specific MOPs should be stated clearly and concisely early in the development phase. For example, all combat models must complete detection and combat processing functions. Reasonable standards based on run-time requirements can be stated for such basic functions. The conditions under which these standards must be met should also be addressed. Performance measures must be quantifiable and capable of supporting detailed analysis.

2. Design

Testability should be an important consideration during system design. Once MOPs are established, the ability to assess the system using these standards becomes critical. For example, taking run-time measurements on key algorithms is an excellent way to gauge computational efficiency and to evaluate the responsiveness of the model. Such testing requirements need to be addressed during the design of the system to produce high quality software.

The management of the game clock is a central design decision for combat simulations. This is particularly true for training systems such as MTWS which must continuously interact with a sizable number of people during exercises. Event synchronous systems ensure the veracity of the game, but may slow the pace of the exercise during peak load periods. This may not be evident to the user since the system controls the rate of advance of the game clock. In contrast, the MTWS version tested during the

Developmental Test was time synchronous. Timing deviations for events were easy to detect rather than hidden, but it was possible for events to be executed out of sequence. Perhaps a hybrid clock management scheme may offer a desirable design alternative. This notion would use a basic event-synchronous design, but would also maintain a separate wall-clock time that would advance at the requested rate. This would provide the means to record any timing lags in game-time while also ensuring the proper ordering of executed events.

Following the Developmental Test, MTWS software was re-engineered to allow the users to choose between the event-synchronous and time-synchronous time management modes. This modification has added significant flexibility to the system. Now the user can decide which scheme best suits their purpose.

3. Testing

The purpose of performance testing is to determine whether the system operates according to pre-determined standards. This requires a rigorous test scenario and the means to collect relevant data. The test scenario must exercise the model according to the standards and conditions of the measures of performance. Once developed, a test scenario should be baselined to conduct comparative studies. Data collection efforts should be specifically tailored to support computation of the MOPs. Automated data collection techniques are generally less expensive and more precise than manual methods in the long run. Therefore, automated testing procedures should be incorporated whenever feasible to support tests over the life cycle of the system.

VI. CONCLUSIONS

The Developmental Test was an important event which directly contributed to the successful completion of the MTWS. The test results highlighted the need improve the overall performance of the system. The most critical finding, that ground moves were more likely to be executed late than other events, was statistically verified.

Based on the detailed analysis of the MTWS Developmental Test, this thesis has offered several recommendations to improve various aspects of performance testing. These suggestions will help ensure that the timeliness and responsiveness of the warfare simulation will meet Marine Corps requirements as new versions of the software are prepared for release. Specific measures of performance were developed and a performance baseline established so that quantifiable comparisons between different MTWS configurations can be made. This will provide a yardstick by which performance improvements can accurately be assessed.

A review of testing procedures highlighted the need to develop automated data collection techniques. Writing essential data to output files will save time and money in the long run, while also improving accuracy. Although this will require additional effort to design and code, the benefits will be accrued over the entire life cycle of the system. Performance measures will be of little value if efficient data collection methods are lacking.

Measures of performance must address both statistical and operational considerations to be thorough and valid. Measures should be based on an interval or higher scale so that more powerful statistical procedures can be employed to gain insight. However, MOPs must also reflect real-world requirements in defining benchmarks for acceptable performance. This will ensure that test results are both meaningful and quantifiable.

The scenario plays a key role in the evaluation of any computer-based warfare simulation. A rigorous test scenario is an essential prerequisite for sound performance tests. In the case of MTWS, a six hour portion of the Developmental Test scenario was found to be suitable for such testing. Enhancements can be made to this segment as necessary to improve performance testing capabilities.

It is hoped that the insight and suggestions contained in this study will prove useful to the MTWS program as it matures during operational use. There are many lessons learned which may also apply to the design and testing of complex warfare simulations in general.

APPENDIX. GRAPHICAL ANALYSIS OF DIFFERENCES IN MULTINOMIAL PROBABILITIES

This graphical procedure is based upon the chi-square statistic formed under the hypotheses that several multinomial distributions are the same. It employs several plots that illustrate different features or components of the chi-square statistic to see whether the distributions are in fact different, and where the differences are. Thus, the chi-square statistic is useful both for formal hypothesis tests and as the basis for graphical analysis to determine if and where the distributions differ, and how severe the differences may be.

This appendix provides detailed theoretical background. It is divided into two sections. Section A explains how a general chi-square statistic can be formed to evaluate the null hypothesis that probabilities of specific outcomes are the same for different multinomial distributions (i.e., that the multinomial distributions are the same). This method is then applied to the hypothesis test of MTWS event distributions in Section B. Together, these sections are intended to lay a better framework for the graphical analysis presented in section C.1 of chapter IV.

A. DEVELOPMENT OF THE CHI-SQUARE STATISTIC

Suppose that data is sampled from $i = 1, 2, \dots, r$ independent multinomial distributions, each with the same set of $j = 1, 2, \dots, c$ possible outcome categories. For distribution i , let N_i be the sample size,

Y_{ij} be the number of outcomes from distribution i in category j ,

p_{ij} be the probability that an outcome from distribution i will be in category j , and

m_{ij} be the expected number of outcomes from distribution i in category j .

Therefore,

$$N_i = \sum_i Y_{ij}, \text{ and}$$

$$\sum_i p_{ij} = 1$$

by definition, and the pooled sample size is

$$N = \sum_i N_i.$$

The c -vector of probabilities for distribution i is denoted by P_i , such that

$$P_i = p_{i1}, p_{i2}, \dots, p_{ic}$$

for each i . If the probabilities (P_1, P_2, \dots, P_r) are unknown, then under the null

hypothesis $H_0 : P_1 = P_2 = \dots = P_r$ the pooled estimate for the probability for outcome

j , is

$$\hat{p}_j = \frac{\sum_i Y_{ij}}{N}$$

for all j . The expected frequency of outcomes from distribution i in category j is

$$m_{ij} = N_i \hat{p}_j$$

for all i and j .

Then,

$$Q = \sum_i \sum_j \frac{(Y_{ij} - m_{ij})^2}{m_{ij}}$$

is asymptotically distributed as a χ^2 random variable with $(r-1)(c-1)$ degrees of freedom.

The difference between Y_{ij} (i.e., observed frequency) and m_{ij} (i.e., expected frequency) is known as the residual. Further, note that

$$\frac{(Y_{ij} - m_{ij})^2}{m_{ij}}$$

is the contribution to Q due to the difference between the observed and expected numbers of outcomes from distribution i in category j . Similarly,

$$\frac{(Y_{ij} - m_{ij})}{\sqrt{m_{ij}}}$$

is referred to as the standardized residual for an outcome from distribution i in category j .

Finally, the relative frequency of distribution i in category j , denoted as f_{ij} is

$$f_{ij} = \frac{Y_{ij}}{N_i}$$

for all combinations of i and j .

B. HYPOTHESIS TEST OF MTWS EVENT DISTRIBUTIONS

In the case of the MTWS event timing data five multinomial distributions (i.e., $i = 1$ to 5) are being compared, one for each event type as defined in Table 18. Each distribution has three possible outcomes (i.e. $j = 1$ to 3) based on the event time differential as summarized in Table 19. Table 20 shows the observed and expected counts for the Developmental Test Event Timing Data.

i	Event Type Distribution
1	Ground Moves
2	Fire Missions
3	Air Missions
4	Ship Moves
5	Ship-to-Shore Moves

Table 18. Event Type Distributions

j	Outcome Category	Event Time Differential
1	On-Time	Either 0 or 1
2	2 Minutes Late	2
3	Greater Than 2 Minutes Late	Greater than 2

Table 19. Event Outcomes

i		j = 1	j = 2	j = 3	N_i
1	y_{ij}	30	12	180	222
	m_{ij}	149	12	61	
2	y_{ij}	264	19	30	313
	m_{ij}	210	17	86	
3	y_{ij}	186	10	5	201
	m_{ij}	135	11	55	
4	y_{ij}	22	0	0	22
	m_{ij}	15	1	6	
5	y_{ij}	23	0	0	23
	m_{ij}	16	1	6	
Expected Probabilities		p _j	0.67	0.05	0.28
					N = 781

Table 20. Observed And Expected Counts Of Developmental Test Timing Data

Then the null hypothesis is that the distributions of all five event types are identical ($H_0: P_1 = P_2 = \dots = P_5$), meaning that all events should be equally likely to be on-time, two minutes late, or greater than two minutes late regardless of event type. Under this hypothesis, the estimated probabilities for each outcome are $p_1 = 0.67$, $p_2 = 0.05$, and $p_3 = 0.28$. However, assuming a χ^2 distribution with eight degrees of freedom, the chi-square statistic $Q = 464.128$ indicates substantial standardized deviations between observed and expected counts. The null hypothesis that all event type distributions are the same is rejected and the alternate hypothesis that at least one is different from the others is accepted.

LIST OF REFERENCES

1. *Amendment to the Required Operational Capability (ROC) for the Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS) System*, p.1, Marine Corps Combat Development Center, Quantico, VA, 26 July 1994.
2. *Test Plan for the Developmental Test of the MTWS System*, p.1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 14 November 1994.
3. *Developmental Test Software Test Report for the MTWS System*, p.28, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 15 December 1994.
4. *Required Operational Capability (ROC) for the MTWS System*, p.7, Marine Corps Combat Development Center, Quantico, VA, 16 March 1990.
5. Blais, Curtis L., "Marine Air Ground Tactical Warfare Simulation," *Proceedings of the 1994 Winter Simulation Conference*, p.839, ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A. F. Seila.
6. Ibid.
7. Ibid., p.841.
8. Ibid., p.840.
9. Test Plan for the Developmental Test of the MTWS System, p.2.
10. Blais, p.841.
11. Ibid., p.840.
12. Test Plan for the Developmental Test of the MTWS System, p.12.
13. Developmental Test Software Test Report for the MTWS System, p. 28.

14. Test Plan for the Developmental Test of the MTWS System, Appendix A.
15. Developmental Test Software Test Report for the MTWS System, p. 6.
16. Test Plan for the Developmental Test of the MTWS System, p. 5.
17. Developmental Test Software Test Report for the MTWS System, p. 7.
18. Required Operational Capability (ROC) for the MTWS System, pp.7-8.
19. Test Plan for the Developmental Test of the MTWS System, p.26.
20. Developmental Test Software Test Report for the MTWS System, p. 32.
21. Ibid., p.10.
22. Ibid., p.6.
23. Test Plan for the Developmental Test of the MTWS System, p.32.
24. Developmental Test Software Test Report for the MTWS System, p. 21.
25. Ibid., pp. 21-30.
26. Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, p.291, McGraw-Hill Inc., 1982.
27. Lewis, Peter A. W., *UEDIT 1.09: An APL2 Input/Output Editor and Exploratory Data Analysis Tool*, Operations Research Department, Naval Postgraduate School, 11 September 1992.
28. Developmental Test Software Test Report for the MTWS System, p. 21.
29. Conover, W. J., *Practical Nonparametric Statistics*, pp. 153-156, John Wiley and Sons, Inc., 1980.

30. Kemple, William G., Lecture Notes, OA3104 Data Analysis, Ch.6, Summer 1994.
31. Chambers, John M. et al, *Graphical Methods for Data Analysis*, pp.21-24, Wadsworth and Brooks / Cole Publishing Company, 1983.
32. Chambers, John M. et al, pp. 94 -123.
33. Blais, Curtis L., e-mail message of 30 May 1995.
34. *MTWS Status Review*, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 13-15 September 1994.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3. Professor William G. Kemple Department of Operations Research, Code OR/Ke Naval Postgraduate School	1
4. Professor Bard Mansager Department of Mathematics, Code MA/Ma Naval Postgraduate School	1
5. Director, Training and Education MCCDC, Code C46 Quantico, Virginia 22134-5027	1
6. Commander, Marine Corps Systems Command Attn: PM Training Systems Quantico, Virginia 22134-5010	1
7. Commanding Officer Marine Corps Tactical Systems Support Activity Attn: MTWS Project Officer Camp Pendleton, California 92055-5000	1
8. Commanding Officer Naval Command and Control and Ocean Surveillance Center Research, Development, Test, and Evaluation Division Attn: Mr. John Chang 53560 Hull Street San Diego, California 92152-5001	1

- | | |
|---|---|
| 9. Mr. Curtis L. Blais
VisiCom Laboratories
10052 Mesa Ridge Court
San Diego, California 92121 | 1 |
| 10. Director, Studies and Analysis Division
MCCDC, Code C45
Quantico, Virginia 22134-5130 | 1 |
| 11. Major William A. Sawyers
Studies and Analysis Division
MCCDC, Code C45
Quantico, Virginia 22134-5130 | 2 |